Amiga: Command Line Interface

The Power of Multiplan for the 128PC

THE Guide TO COMPUTER LIVING

June 1986

$2.50 U.S.

$3.50 Outside U.S.

Vol. No. 3
Issue No. 3
P.D.C.

A Monthly Publication For Commodore™ Owners

(Formerly The Northwest Users Guide)

"The Golden Gate" by Thomas Scott Nelson

06

74470 17515

Explore the Graphics Power of Your 64
With Flexidraw's Light Pen and Software

NEW !

Announcing . . .

SEQUEL

by Bob Richardson

SEQUEL IS NOW ON-LINE
Call 24 Hours a Day
At 300 / 1200 Baud —
See it for yourself!
(503) 659-5752

SEQUEL is finally available! Bob Richardson has taken all of the suggestions from sysops using **Modem Master** and incorporated them into what we feel will become the new standard for Commodore 64 bulletin boards.

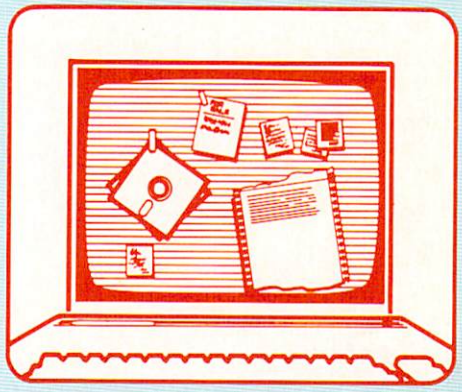
SEQUEL is by far the most powerful and cost effective BBS on the market. The simplicity in design and the ease of use that made **Modem Master** the best-selling board in the country last winter have been combined with a multitude of new features. Offering up to seven message bases, with individual read/write access, **SEQUEL** provides for 33 different combinations of security levels. Message bases may be hidden to provide yet greater security.

Messages are automatically formatted for full right/left justification, and may be read by individual selection. The user is provided with the ability to scan all new messages, messages addressed to the user, or messages written by the user. A new mail forwarding option is also included. Maximum SAVE time for a message on the 1541 is **only 7 seconds!**

E-Mail commands are in a separate menu and allow individual deletion and E-mail forwarding. E-Mail check at sign-on is performed in as little as 1/2 second.

Other features include:

- Fully supported by **both** X-Modem and New Punter (C1) protocols
- Supports up to **four** dual drives
- Download libraries may be organized in up to 99 directories, with as many as 99 files in each directory (depending, of course, on the capacity of the disk drive configuration being used)
- Each directory may have its own user security level, cost (if any), and read/write status
- Directories are stored on the file transfer disk, **not** the data disk so transfer disks may be conveniently changed **while** a user is online!
- A unique Personal File Tranfer system which allows a user to send a program to another individual user



- **Other features:** — system poll/survey; history log; nickname; phone number; remote sysop access; information files; **accurate** clock/calendar; fully automated **unattended** operation; emergency file restoration and the ability to later add user expansion modules

Modems supported by **SEQUEL** include the Commodore 1650, the "old" 1660, the 1200 baud 1670, Mitey Mo, MPP 1064, the Westridge 6420, Hayes Smartmodem, VolksModem, TransCom and many others which are compatible with the ones listed above.

Now available from Aquarian Software

ONLY $39.95

P.O. Box 22184, Portland, OR 97222 — (503) 654-2641

FOR COMMODORE 128 AND C-64 OWNERS ONLY:

BLITZ!™

NOW SHOWING AT SKYLES ELECTRIC WORKS

If your programs are slow, BLITZ!™ them

It's Spectacular
Bob has discovered a spectacular compiler for the **Commodore 64** and the **Commodore 128** that he calls ***BLITZ!***

It's Fast
BLITZ! is faster than *PET SPEED*, faster than any other Commodore compiler that has come down the pike. Your ***BLITZ!***-compiled program will run from *five to 20 times* faster than it did before you blitzed it.

It Translates
This disk program translates your slow BASIC programs into a much, much faster code. The ***BLITZ!*** compiler significantly improves the performance of your BASIC routines. It reads the entire program, reduces that program's size by deciding which operations need to only run once, interprets the operations, and then re-writes the program into its own special P-code.

C-64 or C-128

It's a Treasure
Using this compiler, you can now create a string of BASIC programs such that one automatically loads the next. You can also pass information from one program to another. You will find the ***BLITZ!*** compiler a treasured tool in your programming kit.

It's Popular
Reviewers and users have heaped the following praise on the ***BLITZ!***: "The ***BLITZ*** basic compiler is a dream come true for BASIC programmers." Compute!'s Gazette "worth its weight in gold." TPUG "Wonderful" "Fantastic" "I can't imagine programming without it" User comments. It is even more popular with professional programmers. Many commercial programs now say "***BLITZ!***" when they are listed. The ultimate accolade for ***BLITZ!***.

	SUMMER SALE
BLITZ! C-128, Disk	**69.95***
BLITZ! C-64, Disk	**49.95***

QUICKSILVER 128™ now available for Commodore 128

Is the fastest IEEE-488 Interface available for the **Commodore 128.** How fast? As fast as the external IEEE-488 device will allow. For example, a program or file will load over four times faster from a Commodore 4040 or 8050 disk drive as compared to a 1541 disk drive. ***QUICKSILVER 128*** loads 5 times faster than a serial MSD drive. When used with printers speed is only limited by the printer.

QUICKSILVER 128 is transparent. ***QUICKSILVER 128*** does not occupy any program memory. Almost all programs are completely compatible with ***QUICKSILVER 128.*** A word of caution, some IEEE disk drives have different formats, and 1541 formatted programs are not going to load. Some programs protect the disk by various operations in the 1541 disk memory. These programs will not load on non-1541 disk drives. However, for most uses ***QUICKSILVER 128*** is very quick and transparent.

C-64 or C-128

QUICKSILVER 128 will interconnect your **Commodore 128** to Commodore SFD 1001, 2031, 2040, 3040, 4040, 8050, 8250, 9060, and 9090 Disk Drives. ***QUICKSILVER 128*** will connect your **Commodore 128** to Commodore 2022, 2023, 4022, and 4023 Printers.

QUICKSILVER 128 is easy. Easy to plug into the cartridge port. Easy to plug additional cartridges into the built-in ***QUICKSILVER 128*** cartridge port extender. Easy to locate your IEEE device within 6 feet of ***QUICKSILVER 128*** using the built-in horizontal, cartridge port extender. Easy to select which device numbers you want ***QUICKSILVER 128*** to interface to the IEEE bus. Easy to use with almost all programs. Easy to buy, at an amazingly low price, from Skyles Electrics Works or your local dealer.

	SUMMER SALE
QUICKSILVER 128, C-128 Cartridge	**$119.95**
IEEE Flash! 64, C-64 Cartridge	**99.95**

So many cartridges, so little space . . .

Skyles Electric Works to the rescue, with our 2+1™ and 4+1™ expansion boards for the C-64. Briefly the **2+1** gives you 2 vertical and 1 horizontal fully switchable cartridge ports. The **4+1** continues the tradition with 4 vertical and 1 horizontal cartridge ports. See the inside front cover for more details.

		SUMMER SALE
Skyles 2+1 Expansion Bd. C-64	**$49.95**	**39.95***
Skyles 4+1 Expansion Bd. C-64	**$89.95**	**69.95***

*There is an additional $4.50 U.S. and Canada, $15.00 Europe and Asia, shipping charge per order. California residents add sales tax.

From Skyles Electric Works, the oldest and largest professionals in the business.

This is just a few of the 200+ bargains from the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full splendor, and the other pages overflowing with over 200 bargains in peripherals, software, and books that will make your Commodore 128 or C-64 computer even nicer to live with.

So, if we missed sending you your very own copy within the last few weeks, call us at 1-800/227-9998, unless you live in California, in which case call 1-415/965-1735.

Skyles Electric Works
231E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

THE GUIDE TO COMPUTER LIVING

The cover art, "The Golden Gate", was created by Thomas Scott Nelson with Aegis' **Images** on the Amiga.

The Guide features high quality original artwork on the cover each month. All artists are encouraged to submit their computer artwork for consideration. The only restriction is that the art must have been created using a Commodore computer. This could be your chance to move that masterpiece from the screen on your monitor to the newsstands of America! And make a few dollars in the process.

Please submit all artwork on disk, with a cover letter describing the graphics package used to create it. It will be photographed from the screen, so screen dump capability is *not* a requirement.

The Guide To Computer Living
3808 S.E. Licyntra Ct.
Portland, OR 97222
(503) 654-5603

The Guide is an independent monthly publication serving Commodore computer owners. The views expressed in *The Guide* are those of the individual writers and do not necessarily constitute an endorsement of a product. *The Guide* reserves the right to determine the acceptability of any advertising submitted.

Letters to the editor are not only welcomed, but encouraged, and will be printed as space permits.

The Guide is published monthly (and is available by subscriptions at the following rates: 1 year-$18.00; 2 years-$35.00; 3 years-$48.00) by Aquarian Communications, Inc., 3808 S.E. Licyntra Court, Portland, Oregon 97222. Second class postage paid at Portland, Oregon. POSTMASTER: Please send address changes to *The Guide*, 3808 S.E. Licyntra Court, Portland, Oregon 97222.

TABLE OF CONTENTS

June 1986

COMING NEXT MONTH . . .

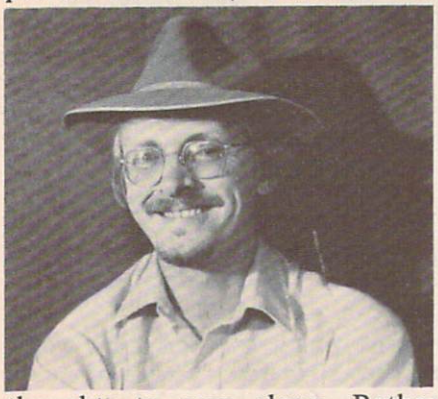
MULTI-TASKING ON THE AMIGA

RND (0) Notes:

Amiga Sales Surpass Atari's 520

by Randy Chase

Well gang, this month these will indeed be random notes. I have a lot of assorted bits and pieces of news, rumors and



thoughts to pass along. Rather than spend a lot of time trying to tie them all together, I think I'll just start wading through them.

Amiga Sales Update

Just before going to press, we received a very interesting press release from InfoCorp, an independent marketing research company, regarding Amiga sales. Their figures show that the Amiga not only outsold the Atari ST for the months of December, January and February, but also in December (traditionally the largest volume month of the year) the Amiga outsold the *entire line* of Atari products.

Considering that these sales figures don't reflect the major increases in consumer activity being reported to us by dealers since the price reduction (or promotional campaign, if you prefer to use Commodore's Newspeak), the Amiga is beginning to carve out a very significant niche for itself.

Interestingly enough, when I called Commodore and tried to get a response to the market figures released by InfoCorp, they didn't have any comment. After being referred to Frank Leonardi, Senior Vice-President of Sales for Business Systems (and supposedly the man responsible for Amiga sales), I was informed by his office that he didn't talk to the press. Rather an unusual policy for someone in his position.

It would seem to me that everyone at Commodore would be wanting to tell *everyone* about the increasing acceptance of the Amiga. After all, this is the computer on which the future of corporate Commodore supposedly rides. Oh well, I guess it is important to remember that this *is* Commodore we are watching, and that in itself means that we shouldn't expect them to act (or react) in a normal or logical manner.

I did find an Amiga distributor that was willing to talk about Amiga sales. He reported that in his region the Amiga was outselling the Atari by a two to one ratio, and that his sales were 163% over his quota. He also stressed that the Amiga price reduction was in fact a "promotion" that would expire in June and prices *would* revert to full retail.

He acknowledged that the price cut has had a "very strong impact" on sales, and I just don't think that Commodore can ignore the highly competitive 520 ST and raise their price once they've dropped it. If anything, I expect that we will see the prices continue to creep downward. I guess we'll just have to wait and see. Anyone back at Commodore have any comments on the subject? We'd be more than willing to listen if anyone wanted to talk.

With the price coming down, there just isn't that much of a dollar difference between the Amiga and the Atari. It's reassuring to know that the consumers are sophisticated enough to see through the negative advertising campaign that Tramiel continues to wage, and are choosing to spend the extra dollars for the Amiga.

I know that I'm starting to sound redundant, but, once again, I feel I must comment on the nature of Atari's advertising campaign. The 520 ST is a nice computer, and a very attractive package for the money. Why can't the people at Atari settle down and try to sell their computers on their own merits? Is it really necessary to continue with these "rip off" ads that seemed directed at downgrading everything else on the market rather than stressing the product they are advertising? Come on Jack, comparative advertising is one thing; but deceptive mud slinging is quite another. In the long run, this kind of advertising doesn't help anyone; and from the sales figures we're seeing, they obviously aren't convincing the public.

RAM Disks Now Available

The biggest hardware development in the Commodore world this month is the arrival of the RAM disks for the 128. The 1700 RAM expansion offers an additional 128K and sells for about $130, while the 1750 offers an additional 512K and is retailing

for $300. The first reaction I have is that $300 to expand a $250 computer seems like a ridiculously high price, especially when you realize that there isn't *any* software currently available that *will* allow you to utilize this expanded memory.

CP/M: Hello and Goodbye?

The one exception to the utilization of the RAM disk, however, is CP/M; which brings us to another interesting subject. I'm still hearing that Commodore is going to be removing the Z-80 chip from the 128 PC and adding more memory instead. Again, Commodore doesn't seem to have much to say, one way or another.

With the lack of interest in the 128 on the part of CP/M publishers, it wouldn't surprise me if Commodore eliminated the chip. Supposedly, the Z-80 was *almost* removed from the machine just prior to its release. Considering the limited market for CP/M software, I really expected the software companies still offering CP/M titles to jump at the opportunity to support a machine which had the potential to breathe new life into what is an almost extinct operating system. Alas, they obviously would rather continue to try to sell over-priced and outdated software to Osborne orphans than to jump into the highly competitive Commodore market.

128 News and Speculation

Unconfirmed rumors (are they any other kind?) still say that Commodore will replace the Z-80 chip with more memory, and may even offer the 128 with a built-in 3½ inch disk drive. It would be nice if they would make up their minds, and let us know one way or another. I hear so little from users about CP/M that I wonder if anyone would really care one way or another. Perhaps if there were a little more support from the CP/M software publishers, and the manuals were readily available, users would be a little more enthusiastic.

The 128 mouse is finally available, although the only software I know of that makes use of it is **Jane**, and from what I've seen of her, I wouldn't recommend bothering. The other thought that warrants mention, while we're talking about mice, is that if you buy a 128, add a 1571 disk drive, a mouse, and memory expansion, aren't you somewhere in the financial neighborhood of the Amiga?

Of course, you do have the option of not having to run your 64 software on your 64; you can instead run it on your 128 in the 64 mode. Maybe I've missed something, but it seems to me that the 128 needs some significant pricing adjustments to bring it into line with the computing value offered for the dollar by both Commodore and Atari.

Having had a chance to experiment some with the MS-DOS software emulator for the Amiga, I confess I wasn't very impressed. Perhaps I'm spoiled, having spent my whole computer life working with colorful Commodores, but I just can't get that excited about taking a machine as amazing and versatile as the Amiga and transforming it into just another clone. However, if you want a tantalizing idea to play with, think for a moment about taking your 128, adding the 512K expansion, and plugging in an MS-DOS cartridge. Sound interesting? Sure does, doesn't it?

In January last year, when they first showed the 128 at the winter CES in Las Vegas, one of Commodore's engineers very excitedly told me about the MS-DOS cartridge they had developed for the 128 (reportedly containing an 8088 chip). Not only was this machine going to run *all* C-64 software and read *most* CP/M formats, it was going to also have the ability to emulate an IBM.

Six months later, in Chicago, when I inquired about the MS-DOS cartridge, no one seemed to have any recollection of the conversations I clearly remembered from Las Vegas. One less timid (and obviously un-named) source said that with the forthcoming release of both the Amiga and the PC 10 (Commodore's IBM clone that is marketed quite successfully in Canada) they had decided not to allow the 128 the ability to compete directly with these bigger and more profitable machines.

Well, here we are, getting ready to go to Chicago again, and we still *can't* buy a PC 10 in the

The Guide Staff

Publisher & Managing Editor
Randy Chase

Associate Editors
Grant Johnson
Bill Wallan

Contributing Editor
Mindy Skelton
Robert J. Sodaro

Staff Artists
Eddie Johnson

Photographs
Grant Johnson

Production
Guion Mawr
Bill Wallan

Advertising Sales
Bronwen Fitzhugh
Jim Losk
(503) 297-3541

Staff Writers
Lyn Chase
Dr. Petrie Curryfavor
Bud Izzit
Eddie Johnson
John R. Olsen, Jr.
Bob Richardson
Shelly Roberts

Contributors
Kelly Allen
Carmen Artino
Valerie Jean Kramer
Chris Scott

Distributed exclusively by
Publishers Distributing Corp.
P.O. Box 8991
Malibu, CA 90265

states, the hardware emulation for the Amiga still *isn't* available (and the software emulation *is* too slow!), and with the changes in the market place, the 128 is no longer the versatile price breakthrough in home computers that it seemed to be back then. It would appear to be time to either drop the price on the 128 (perhaps replacing the 64 as the entry level machine) or find a way to give it some serious enhancements and justify its comparatively hefty price tag.

A 512K RAM disk with no software to utilize it isn't going to be the answer. Perhaps Commodore should look at ways to redefine both the pricing and market niche of the 128. Granted, sales are still strong, but how long are people going to pay *too* much for it just because it *will* run their 64 software?

Time will only tell what Commodore has up their sleeves, or on their drawing boards. Lord knows I've long since quit trying to predict their actions, and they've certainly not shown any inclination of tipping their hand. With Commodore participating in both COMDEX and CES, we should in the next few weeks get some serious indications of what directions they plan to move. We hope that by next month we'll have something real to talk about. Maybe a new machine? (Or at least new in the states!) Maybe new and more competitive prices? And most likely, at least one redesigned and repackaged machine.

Until next month, have fun treading on the rumor mill. And remember, with our dear friends back in PA, there really aren't any speculative tales too tall not to ponder for at least a moment or two. After all, this is a company in which the V-P of Sales doesn't offer public comments. Who knows what lurks behind that mysterious fog of silence that seems to engulf West Chester?

■

Multiplan Meets the 128 PC
Faster, Better, And Still the Same

by Grant Johnson

Sitting in front of a screen full of numbers for hours on end doesn't sound like much fun, but hundreds of thousands of people choose to do just that every day. It seems improbable, yet as you watch, it becomes obvious that the numbers add great meaning — even drama — to the lives of these otherwise ordinary people. All this numerical intensity is generated by a kind of program that, more than any other, made the personal computer a fixture in American life.

The Spreadsheet

Traditionally, a spreadsheet was a tool of the accountant or actuary. It was literally a large sheet of columnar (ledger) paper divided across the top into columns and down the side into rows. In a typical application, the columns might be the months of the year and the rows represent things such as "Cost" and "Quantity". The intersection of the "June" column with the "Cost" row would represent June's costs.

Although the same information was probably to be found in a journal or daily ledger elsewhere, the spreadsheet, being a condensed summary, made it possible to easily spot important events and trends that might otherwise go unnoticed. It would be obvious, for example, that costs had risen the last four months out of five.

If you visualize Bob Cratchett, quill in hand, slavishly making entries under Scrooge's cold stare, then you have the idea. As valuable as they are, spreadsheets are a lot of work.

It may be the military that pioneers computational technology, but it is business that domesticates it, and, from the era of the tabulating machine right to the present, business has lavished the best it could afford on the columnar summary. Often, the full resources of giant accounting systems were focused on the production of a few key sheets of paper — the cap stone on a pyramid. And you can bet those pages weren't left about as casual reading for the kid in the mail room. No effort was spared in keeping the results strictly factual. Any speculation about alternatives were called "simulation" or "modeling" and were the province of specialists.

Cratchett's Revenge

Then, in 1979, Robert Frankston and Dan Bricklin decided to use microcomputers to take some of the drudgery out of the spreadsheet. The idea was to make the numbers as easy to enter, use and update as the words are in a wordprocessor.

Their original product was VisiCalc, and it was such a success that many people bought an Apple II computer just so they could use VisiCalc! It was great. You no longer needed to be a programmer, or even have access to a main frame computer, in order perform these basic accounting functions. You got *immediate* results. You could personalize the thing to meet your needs, and you could make changes in the easy, interactive fashion we all now expect from our micros.

The upshot of all this was that you no longer had to inhabit a board room to see a spreadsheet. Even the kid in the mail room could afford his own. Suffice it to say, Frankston and Bricklin had found a better mouse trap and the world beat a six-lane road to their door.

Fatal Flattery

If imitation can be said to be flattery, VisiCalc's creators should never want for a kind word in this life. Copies, clones, "improvements", adaptations and extensions proliferated. Whatever can be said for the "me too's", they all contain the essence of VisiCalc. The competition nibbled away at the edges of VisiCalc's terf until Lotus 123 took the high ground.

Yet even the latest generation of large programs pay homage every time they appear on a screen. The second generation of spreadsheets may be integrated with graphics and even have data base functions, but they tend to be more complicated and run only on the big-bucks machines.

Microsoft's **Multiplan**, at least the version available for the 64 and 128 PC, is of the first generation type — strictly a spreadsheet. But it is a very good spreadsheet.

How It Works

Frankston and Bricklin organized their creation into "cells"; each of these cells corresponds to the intersection of a row (horizontal) and a column (vertical). In **Multiplan**, the rows

and columns are numbered from the top left-hand corner of the screen. The cell in this upper left position is Row 1, Column 1 or "R1C1" for short. As you can see below, in **Multiplan** they are numbered in "reading" order. (Many spreadsheets use letters of the alphabet for column designation, a practice that I prefer.) The "R & C" notation used here may be thought of as coordinates or cell addresses.

```
     1      2      3      4

1  R1C1   R1C2   R1C3   ...
2  R2C1   R2C2   R2C3
3  R3C1   R3C2   R3C3
4  ...
```

There are four basic types of information that may be associated with a given cell in **Multiplan**: Alpha (descriptive text), Value (numbers), Labels (names used as an alternate way to address a cell) and Formulas (more about which shortly).

A page or screen of numbers is not particularly meaningful by itself, so **Multiplan** allows you to enter natural language stuff such as "TOTAL" and "COST" to help it all make sense. Value entries are numbers, and will change as rapidly as the circumstances pertaining to them. This versatility is one of the strengths of the spreadsheet.

The values of "TOTAL" and "COST" will also change, but the method of calculating them (the Formula) will not. If you are a programmer, you may recognize them as "constants", and if your background is in mathematics you are welcome to think of them as "givens".

```
     1          2

1  MATERIALS   $ 55.00
2  LABOR       $ 45.00
3              -------
4  TOTAL       $100.00
```

MATERIALS, LABOR, TOTAL and even "--------" are just text that describe the nature of the numbers next to them. $55.00 and $45.00 are value entries, but their sum ($100.00) was supplied by the spreadsheet. How did the program know that MATERIALS and LABOR were to be added together and where the results were to appear?

A formula was entered in row 4, column 2 (R4C2). In addition to the optional row and column numbers that we have been using thus far, **Multiplan** also prints out a spreadsheet that includes the "behind-the-scenes" information (such as formulas).

```
     1          2

1  MATERIALS      55
2  LABOR          45
3                 ---------
4  TOTAL          R1C2+R2C2
```

Here we see the formula behind our total. In "English" it simply tells **Multiplan** to take the value found in row 1 column 2, add it to the value at row 2 column 2 and place the sum in row 4 column 2 (which is where the formula itself resides). The nice thing about our formula is that when MATERIALS and LABOR change, TOTAL will always reflect their true sum. **Multiplan**'s ability to display the formulas used within it can be very important when you need to verify or modify a large sheet.

While we are poking around behind the scenes, there are several other points to be made. The entries that the **Multiplan** literature refers to as "Alpha" can be seen here for what they are — text literals. They have no meaning to **Multiplan** beyond the fact that the user wants them to appear as, and where they are.

Multiplan has many "formatting" options as well. Our examples have employed two of them. In making the above printouts, **Multiplan** was told that I wanted all alpha information to appear centered within its cell and that all values (even those created by formulas) were dollar amounts with two decimal places. The floating dollar sign, decimal point placement and rounding of digits

```
    1          2         3         4         5         6         7
1                  January   February  March     April     May       June
2
3  Sales           $20000    $20000    $20000    $20000    $20000    $20000
4
5  Cost
6       Material   $6000     $6000     $6000     $6000     $6000     $6000
7         Labor    $7000     $7000     $7000     $7000     $7000     $7000
8      Overhead    $4000     $4000     $4000     $4000     $4000     $4000
9                  --------
10    Total Costs  $17000    $17000    $17000    $17000    $17000    $17000
11
12
13
14
15 Gross Profits   $3000     $3000     $3000     $3000     $3000     $3000
16
17
18
19
20
COMMAND: Alpha Blank Copy Delete Edit Format Goto Help Insert Lock Move
         Name Options Print Quit Sort Transfer Value Window Xternal
Select option or type command letter
R18C2   R[-4]C+R[-3]C+R[-2]C          97% Free      Multiplan: SPENCER
```

Eighty-column display is a real boost on the 128 PC. You even get an extra row.

beyond two places are all automatic from that point on.

Multiplan has numerous ways of addressing its cells. I mentioned earlier that it was possible to assign labels to a cell (or group of cells). In the following sheet, rows one and two of column two have been assigned the labels MATERIALS and LABOR, respectively. It is merely good spreadsheet practice to use the text labels (seen in column one), but I could have used anything I wished. Note the clarity of meaning in the formula of row four.

```
      1              2
1 MATERIALS          55
2 LABOR              45
3              ------------
4 TOTAL MATERIALS+LABOR
```

Multiplan also offers "relative" addressing of cells. The TOTAL row formula could also be written as:

```
4 TOTAL      R[-3]C+R[-2]C
```

In this sheet fragment the cell containing the MATERIALS value (55) is referred to by `R[-3]C`. **Multiplan** understands this to be the present row (where the formula is located) minus three rows. A "C" by itself simply mean the current row ("R" by itself would likewise refer to the present row). Starting from the formula cell, look up three rows in the current column and you will find the 55 needed. At `R[-2]`, the LABOR value (45) is just below it. Relative addresses are especially valuable when a formula is to be used again in, say, another column.

```
      1              2              3

1 MATERIALS          55             65
2 LABOR              45             50
3              ------------ ------------
4 TOTAL    R[-3]C+R[-2]C  R[-3]C+R[-2]C
```

The exact formula that gave us the total in column two now gives us the correct total in column three. **Multiplan** has editing facilities that make duplicating an existing formula quick and error-free.

Lest you think that you will go through life with calluses on your "R" and "C" fingers, **Multiplan** will usually supply cell coordinates for you. All you have to do is move the cell pointer to the location you wish to indicate.

A discussion of **MultiPlan** wouldn't be complete without at least a mention of **Sideways** from Timeworks. This novel utility works with not only **Multiplan**, but also most of the popular spreadsheets for the Commodore 64 and 128 computers.

What does **Sideways** do, you ask? Just what it implies. It prints *sideways*. It works with most popular dot matrix printers that support dot addressable graphics and will print your report out sideways on the paper, allowing you to have one long continuous vertical page.

Sideways will work with any Commodore spreadsheet that will save the data to disk in an ASCII file. It also allows selection of various print sizes, line spacing, top and bottom margins, and even provides for a double-strike option to improve the appearance of that business-related sheet.

Anyone who is seriously using their spreadsheet should give serious consideration to this practical and versatile program.

Control Facilities

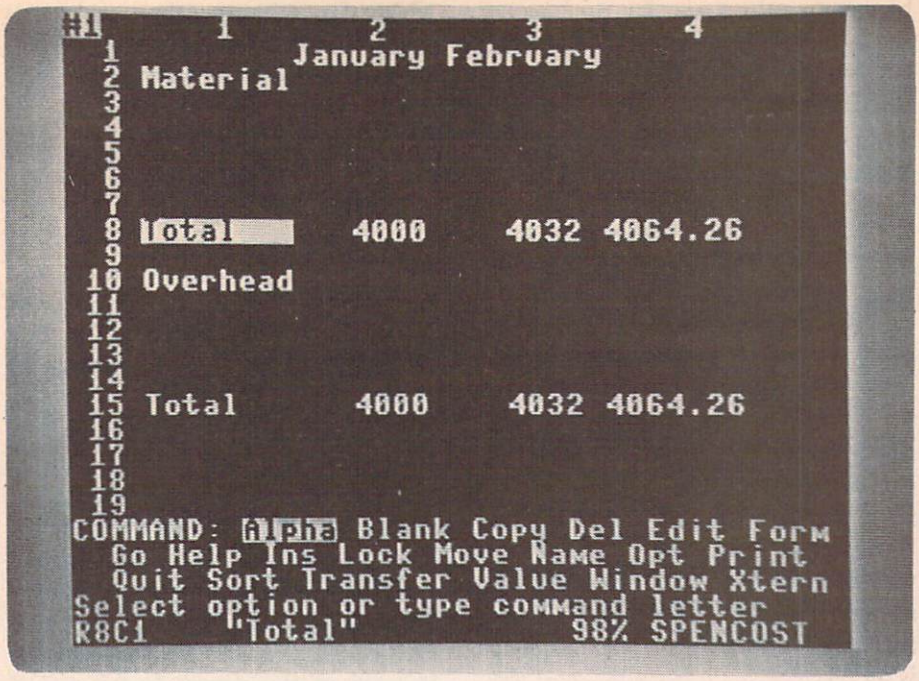
Multiplan's screen consists of two major parts; the spreadsheet proper and, at the bottom of the screen, a command menu and status lines. The spreadsheet has row and column numbers arrayed across the top and down the left edge. For a fresh sheet, the rest of this area is blank except for a reversed video bar called a cursor pointer. This pointer is an indicator of the active cell; *i.e.*, the cell into which values, formulas and the like will be put. The position of the pointer is echoed at the bottom of the screen in "R & C" notation.

The cell pointer behaves much like a normal cursor, and is, in fact, controlled by the cursor keys. There are 20 rows and 7 columns visible (5 columns on a 40-column screen). The size of a **Multiplan** spreadsheet is 255 rows and 64 columns, and the pointer can be used to make the unseen parts of the sheet visible. When the pointer encounters the edge of the screen, the spreadsheet portion of the screen is made to scroll.

There is also another cursor visible in the Command area of the screen. Twenty commands are listed and one of them is highlighted by this "edit" cursor. The TAB key may be used to move this highlight along to the command of your choice. Choice can then be indicated by pressing the return key, or the whole cursor business can be by-passed by simply typing the first letter of the command — "C" for COPY.

The commands are as follows:

- ALPHA is the command used to enter text literals into a cell.
- BLANK is used to tell **Multiplan** that you do not wish a cell, or group of cells to appear.



Typical Multiplan 40-column display. Rows and columns are numbered. Command menu appears at the bottom.

Such blank cells might contain intermediate steps in a complex calulation.

- COPY is my favorite command. If we had been computing the MATERIALS, LABOR and TOTAL for twelve months of the year, all that would have been needed after the first formula would have been to copy the calculation column (2) to the left eleven more times. One use of the COPY command would have done the job.
- DELETE is used to delete cells or even whole columns or rows.
- EDIT is used to view or change the contents of cells. If you need to modify a formula, it can be done in the edit mode. You may move about the formula one character at a time, delete and insert characters, etc.
- FORMAT brings up a series of submenus through which you can control the display of text (centered, right and left justified) and numeric values. Numeric formatting is especially rich. Decimal control, floating currency signs and scientific notation are just a few of the options.
- GOTO is the optional method of getting around the spreadsheet screen.
- HELP — **Multiplan** has extensive help files. This command brings them to the screen for you to read when you need them.
- INSERT is the inverse of delete. With it you may insert whole columns or rows. By the way, the cell references and formulas already in use are modified for you when you add or delete portions of the sheet.
- LOCK is the command to use after you have your spreadsheet working the way you want it. Cells can be locked to prevent their contents from being changed inadvertently.
- MOVE is used to rearrange cells or groups of cells within the sheet.
- NAME — This is the command used to label cells.
- OPTIONS calls up a submenu that allows you to change from auto recalculation to manual. Ordinarily each time a cell is changed **Multiplan** recalculates the entire spreadsheet. On a large sheet the time required can become considerable, slowing data entry. Recalculation can be accomplished at any time by pressing the "recalc" key ("!"). Another important option, the iteration option, enables **Multiplan** to recalculate a sheet continually until some final event occurs. More about that later.

- PRINT, as the name implies, commands **Multiplan** to send data to an output device. It has its own options submenu as well. Printer set up (not well supported) for things like margins and page length are available, as are such choices as the setting of row and column number headings and the option to print the formulas associated with the cells. (In case you are wondering how you can make a paper copy of a spreadsheet that may be more than 640 characters wide and 255 lines tall, **Multiplan** is very clever about printing the sheet in sections that can then be assembled, with perhaps a bit of tape, into a hard copy of your masterpiece.)

PRINT also lets you easily send your output to a sequential file on disk. The sample spreadsheets seen throughout this review were done in this way for later access by a word processor.

- QUIT exits **Multiplan**.
- SORTs can be accomplished within a spreadsheet. It's fun to watch **Multiplan** rearrange an entire sheet in the order of a group of cells. This can be helpful in making the information in a sheet easier to use. You can print, the results, but the changes are not permanent, so don't try to save them.
- TRANSFER is the all purpose command used for loading, saving, deleting and renameing work

sheets on disk. (But I have never been able to format a new disk from within the program — you must plan ahead!) There are options here, too. You may swap disk drives, or you may save your work in one of two formats: Normal and Symbolic.

Multiplan is a standardized program. The Symbolic format refers to an exchange method called SYLK (Symbolic Link). Spreadsheets saved in this format can be read by any other program or computer which supports it. That means that you could send a SYLK sheet by modem to an unknown brand of computer anywhere and be sure that, if **Multiplan** runs on it, it can be used.

Loads can be done in three formats; the two mentioned already and "Other" which, for the most part, means Visicalc and clones.

- VALUE can be used to enter numeric data, although **Multiplan** accepts them by default whenever the main menu is available.
- WINDOW can be used to, among other things, split off column and row text from the rest of the sheet. Thus, you may be making entries in column 20 and still be able to see the text titles from column 1. Also, you may be interested in the results happening in one part of a sheet while you are making entries in a distant area — windowing lets you do both.
- XTERNAL (spelled that way because "E" is already in use for "EDIT") brings us to an interesting difference among spreadsheets. This command is used by **Multiplan** to draw data from other work sheets. You might think that 16,320 cells is more than you'll ever need, and you may be right. But serious speadsheet users (and this *is* a serious program) can often fill them all in a single application! It's really not that hard to do.

Most other spreadsheet manufacturers try to solve the problem by making the dimensions of their sheets very large (8,000 rows is typical). The truth is that such size is mostly theoretical, even on a $5000.00 machine, as there simply is not enough memory in the machine. Other drawbacks with this approach: large sheets recalulate slowly, and errors can be very, very hard to find.

Multiplan's solution is to allow you to break up a large sheet into smaller units. An application might have "Sales", "Production" and "Inventory" support sheets, information from all of which might come together in a "Profit & Loss" sheet. All of the virtues of structured programming apply to this approach, and the truly usable size of the application is as large as disk storage. And the whole thing can run on a Commodore 64!

Functions

Calculation is what a spreadsheet is all about, and to most people nowadays, that often means the ubiquitous pocket calculator.

Like a calulator, **Multiplan** has functions such as SQRT (square root), SIN, LN (natural logarithm) and STDEV (standard deviation). It knows about more than dollars. I once used **Multiplan**'s raw number crunching power to design a boat after my shakey calculus failed me.

Solving simultaneous systems of equations is tricky but entirely possible. Spreadsheets shine when it comes to finding the maximums (MAX) and minimums (MIN) of linear systems problems. These are the problems such as producing an animal feed by using the least costly methods and ingredients while maintaining certain minimum nutritional standards. Heavy stuff? **Multiplan** is ready when you are.

Some of **Multiplan**'s functions even provide the user with the rudiments of a programming language (IF, INDEX, DELTA, AND, NOT, OR, TRUE and FALSE). With them you can make automated decisions and solve problems that require iterative approximations.

Animation

No discussion of spreadsheets would be complete without some mention of "what if". By all reports Frankston and Bricklin themselves were surprised by the way people used their original product. Plugging in factual data is only a small part of the way that people actually use spreadsheets. Once you get the facts on the screen, it seems to be a nearly irresistible attraction to make a change here and there just to see what would happen if ... The numbers on the screen come alive and act out scenarios — possible realities — in much the same way that a series of static drawings can be animated when seen in rapid succession.

When loaded with a spreadsheet such as **Multiplan**, the personal computer exhibits one of its most persuasive virtues; it is an intelligence amplifier.

Conclusion

Multiplan is a first-rate spreadsheet program. It is available in a 64/128 package from Epyx for the suggested price of $59.95. The C-64 version has all of the features of standard **Multiplan** but there isn't room in memory for both the spreadsheet data and the program. Consequently, operation of **Multiplan** on the C-64 is slowed by repeated disk access time as the needed parts of the program are swapped into the computer. It all fits in the C-128, so if you are able to use the 80-column mode, it operates comparitively quickly. Another advantage of the 128 product is that it comes with a fast load booter.

The product comes with an excellent manual (this also seems to be a part of Microsoft's standard). Speaking of standarization, once you learn to operate

Multiplan on one machine, you can operate it on *any* other. The only changes Microsoft allows is for substitution of missing keys. Of course, the flip side of this is that the program doesn't take advantage of a given machines abilities. The original **Multiplan** ran on machines without color displays, so you must accept the white characters on black that these programs provide.

The venerable **Multiplan** is surprisingly contemporary after all these years (it must be remembered that computer years are even shorter than dog years). Its reincarnation on the 128 is a joy, and I recommend this program to anyone with a business to run or numbers to crunch.

Beginner's Corner
Creating Windows on the 128

by Mindy Skelton

As you may have gathered from past articles, I like my C-128. Even though software manufacturers have not supported it in the way I might have hoped (where, oh where is the *ultimate* adventure game I dream of?), the 128 offers new and exciting capabilities that expand both programming and computing horizons.

One of the capabilities that I find most useful and flexible (and the one that, strangely enough, I'm writing about this month) is the ability to create "windows" on your 40- or 80-column screen.

You've probably heard of windows, and if you've seen many of the new 128 programs, you've probably seen their use. When a window is in use, part of the screen is partitioned off, and, for at least a while, the computer acts as if the area contained within the window were the entire screen. All the activity takes place within the partitioned area, without disturbing anything that happens to be on the rest of the screen (unless, of course, you write over it ... more on that later). Now you know the secret (or at least a little of the secret) of the popular "pull-down-menu".

Commodore must have liked the idea of windows, they provided two ways of creating them; one in direct mode, and another via a command built into Basic 7.0. First, we're going to look at how to create a window from direct mode, using the ESCape keys. To create a window in direct mode you need follow only a few simple steps.

For our demonstration, clear your screen, then move your cursor to the place you want to be the top left corner of your window. Now press the ESCape key, release it, and press T. Third, move your cursor to the place you want the lower right corner of your window. Finally, press ESCape again, release it and press B. You have now created a window.

Wasn't that fun? Oh ... you say you can't see the window? Just type something in or call up a directory and see what happens, and where it happens. All the action is in your window (I hope). Want your screen back to the way it was? Simplicity itself ... merely press the CLEAR/HOME key twice, and the screen is restored to its original size. By the way, pressing the shifted CLEAR/HOME key will clear the contents of your window and return your cursor to the upper left-hand corner of your windowed screen.

As convenient as this method of creating windows might be, you've probably already noticed that it might be a little hard to use from inside a program. If you want to create a window in a program you're writing, you have two choices. You can give the character string (CHR$) representation of the ESCape key and cursor control keys to simulate the positioning you use in direct mode.

For example, the following two pieces of code will create a window (on your 80-column screen) in the upper right hand corner of your computer screen:

```
10 PRINT CHR$(147)
20 FOR I=1 TO 79:A$=A$+"]":NEXT
30 FOR I=1 TO 5:B$=B$+"[CUR-
SOR DOWN]":NEXT
40 PRINT LEFT$(A$,40)CHR$(27)"T"
50 PRINT B$
60 PRINT LEFT$(A$,79)CHR$(27)"B"
70 PRINT CHR$(19)
80 FOR I=1 TO 600:PRINT "A";:NEXT
```

```
10 PRINT CHR$(147)
20 PRINT "CUR-
SOR RIGHT (50 TIMES)"CHR$(27)"T"
30 PRINT "CURSOR DOWN (5 TIMES)"
40 PRINT "CUR-
SOR RIGHT (79 TIMES)"CHR$(27)"B"
```

Well, it works, but it's a bit cumbersome isn't it? There is an easier way. Built into BASIC 7.0 is the WINDOW command. The syntax for this command is:

```
WINDOW X1, Y1, X2, Y2, CLEAR
```

This command asks you to supply the coordinates for

the column and where you want the upper left-hand corner, the column and row where you want the lower right corner, and a clear flag to be set on or off. For example:

```
WINDOW 10,5,20,15,1
```

would create a window with the upper left corner in the 10th column over and the 5th row down, and the bottom right corner in the 20th column over and the 15th row down. The number one in the final position indicates that the window area is to be cleared as soon as the window is created. A zero in that position would have left the screen area as it was when the window was created.

One thing I find particularly useful in debugging a program is to include a line in the program (after the end of the program) which creates a window off to the side and automatically runs the program. Something like this:

```
1000 END
1010 WINDOW 50,0,79,20,1
1020 RUN
1030 END
```

With this code in my programs, I can periodically run this line and see how the program is progressing without disturbing my program listing. I can then go back, make needed changes, rerun the window routine, and still have access to the listing.

There is a caution in using windows. While it is true that using a window does not disturb the rest of the screen, anything UNDER the window is lost. In order to restore your screen to its original condition in such a situation, you have to use a machine language routine to store the screen information in a buffered area, and restore it after the window is removed. Be careful, it's not quite a beginner's trick.

A couple of last words. You can create multiple windows in any program, but only the last one you've created will be active. If you wish to utilize a previous window, you merely issue the window command again for the coordinates you want.

Here's one last little sample of windowing. Have fun, and, until next time, happy programming.

Note: To get control of your computer back after this program runs, just press any key.

```
10 PRINT CHR$(147)
100 WINDOW 0,0,20,10,1:X=1:GOSUB 1020
200 WINDOW 59,0,79,10,1:X=2:GOSUB 1020
300 WINDOW 5,5,25,15,1:X=3:GOSUB 1020
400 WINDOW 54,5,74,15,1:X=4:GOSUB 1020
500 WINDOW 10,10,30,20,1:X=5:GOSUB 1020
600 WINDOW 49,10,69,20,1:X=6:GOSUB 1020
700 WINDOW 15,15,35,24,1:X=7:GOSUB 1020
800 WINDOW 44,15,65,24,1:X=8:GOSUB 1020
900 WINDOW 30,0,51,5,1
910 PRINT  "AND THIS IS YET"
920 PRINT  "ANOTHER WINDOW WHERE"
930 PRINT  "ALL THE TEXT IS GOING"
940 PRINT  "TO APPEAR.          "
950 PRINT  "EVEN IF IT TAKES TOO "
960 FOR I=1 TO 500:NEXT
970 PRINT  "MUCH ROOM TO PRINT IT"
980 PRINT  "ALL AT ONE TIME.    "
990 FOR X=1 TO 500:NEXT:PRINT CHR$(147)"
    NOW WATCH!":FOR X=1TO500:NEXT:PRINT
CHR$(147):FOR I=1 TO 503:I=I+1:PRINT I;:
NEXT
1000 GET X$:IF X$=""GOTO 1000
1010 END
1020 PRINT "[SHIFT O][CMD Y(19 TIMES][SH
IFTP])"
1030 FOR I=1TO7
1035 IF I=4 GOTO 1080
1040 PRINT "[CMD Y][SPACE (19 TIMES)][CM
D M]"
1050 NEXT
1060 PRINT "[SHIFT L][CMD P (19 TIMES)][
SHIFT @]"
1070 RETURN
1080 PRINT"[CMD G][SPACE]THIS IS WINDOW"
;X;"[SPACE][CMD M]"
1090 GOTO 1050
```

Crossword Magic:
Be "Puzzled", Not Confused (Or, Come In From The Lea)

by Bill Wallan

Have you ever wondered (however briefly?) how people who write crossword puzzles manage to fit all those words together properly? (Or, even improperly for that matter!) Well, I have ... back in the days when I had little else to think about. It may seem to some that I could have better spent my time wondering why a steam iron comes with a "Permanent Press" setting (??!!), but, fortunately, *that* question didn't occur to me until much later in life, after I had learned that too much wondering can be hazardous to your credibility.

In any case, all my wondering produced not an inkling as to how crosswords are conceived. In one of my less lucid moments, I even attempted to create one, just to see if I was as smart as Mother had hoped. Mother would have been sorely disappointed, had she learned the results. Let it suffice to say that the experience left me convinced that crossword writers simply invent the words to suit the puzzle.

Working the puzzles often seems to confirm that theory, too. Nearly every puzzle I have worked has contained at least *one* word that I have never run across, save in another crossword puzzle. (Admit it, puzzlers! You know I'm right.) For instance, have you ever known someone to speak of going for a brisk walk through the lea? What about the ever-popular "Man's name" clue that inevitably forces you to play down, when you really need to play across. (The kicker, where proper names are concerned, is that, if you have the time, you can document just about any spelling of any name by diligently searching through enough material.)

When proper names fail to fill the void, the authors further perplex us with acronyms, followed by abbreviations of obscure social organizations having a membership of 33⅓ or more.

Were it not for these stumbling blocks, working crossword puzzles would appeal to a yet wider cross-section of people. You must admit that we all need mental exercise on occasion.

It occurs to me that trivia buffs should be interested in such a genre. Take all that "useless" information occupying blocks and blocks of MIM (Man's Involuntary Memory), put it into puzzles, make copies and — *Eureka!* — a trivia contest for your next get-together. Provided, of course, you can put it into puzzles.

The people at Mindscape have marketed a program called **Crossword Magic™**, by L&S Computerware, as a part of Mindscape's **Alert™** line of software. The line's goal is to produce software that combines the best in entertainment and educational value. **Crossword Magic** can help you turn out puzzle after puzzle, each in its turn completely free of leas and Kristofers and the IWQ-QQ. *BUT* ... before we go any further with this review, let's get one thing cleared up ...

Crossword Magic is technically a misnomer, at least if you define crossword puzzles in the classic sense. After all, the program is most certainly not magic (obviously?), and the puzzles you can create with it more closely resemble what Dell has been publishing for decades under the category of "Kriss-Cross" puzzles. **Crossword Magic**'s puzzles, however, have the added feature of numbered boxes at the beginning of the words, providing for the addition of "clues", in the manner of classic crossword puzzles.

Unless you use the manual mode to force entry of words, the program will *not* place words in adjacent rows or columns, as they appear in classic crossword puzzles. The reason for this is that if you have words in the first and second columns, reading down, they most likely will *not* form words going across. The program is written to enter only *complete* words, from the choices you type in.

The only way this writer has found to create a classic crossword puzzle with this program is for the user to do all of the puzzle creating beforehand, load the program, enter the "Manual Mode" and *force* the words into the proper places. You can enter all the words necessary to form solid blocks of puzzle into the unused word file, but the program *still* won't enter them into adjacent columns or rows.

If you wish to accomplish

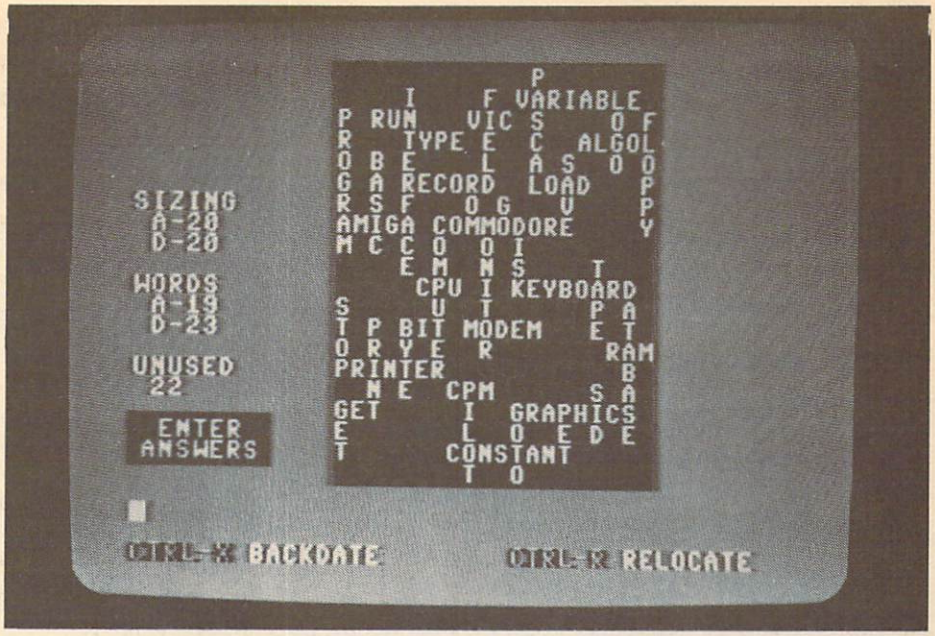
solid blocks, you have no choice but to type the words into the puzzle grid manually, receiving no assistance whatever from the program in the placement of words. And, even then, the program does not necessarily recognize that all of the words formed are present. You could create the same crossword puzzle using only a piece of notebook paper, a pencil and a lamp by which to see.

When working "Kriss-Cross" puzzles, the puzzler is provided a list of words, listed alphabetically by word length, that fit into the puzzle grid. "Clues", per se, are not required, since the meaning of the word is not necessary to the user in order to fit it into the puzzle — only the word length and correct spelling.

Kriss-Cross puzzles have provided many people with many hours of puzzle enjoyment, and therefore must be considered a valid game. And, **Crossword Magic**'s puzzles have the added option of using definition clues, to give the feel of a crossword to the puzzle. In view of these facts, my complaint boils down to an argument over semantics, (nearly always a waste of time). To belabor the point would not be doing justice to a well-written program. So, now that I have that off my chest, perhaps we can get on with this review . . .

For those of you who are interested in creating and/or playing this kind of puzzle, **Crossword Magic** is, indeed, a helpful little tool. You no longer need be plagued by obscure or archaic words and phrases that tend to clutter up commercially produced puzzles, rendering them boring instead of challenging. Now you can provide hours of puzzle fun for your friends and family, or simply store the puzzles away to work yourself once memory has faded into renewed interest.

You choose the topic or theme. *You* make the word list.



The computer assists you in positioning the words to form a puzzle. If you end up with obscure "un-words" (speaking of such), it's because you *chose* to when making the word list.

Crossword Magic comes with a carefully prepared, easy-to-read manual that guides you step-by-step through the process. The disk-based program provides a menu at the bottom of most screens, showing the options you have. The commands required to choose the options, however, are not always obvious, so you will want to keep the comprehensive manual handy for occasional reference.

The actual creating process is simply a matter of typing words into the computer. The puzzle is built one word at a time, (unless you choose to fill the unused word list with many entries and enter them all at once). You type a word and enter it. The computer puts it into the puzzle grid where it will fit. The "Relocate" command will have the machine check for other places the word (or words) will fit, and, if any are found, each is displayed at your command, until all locations the computer finds have been shown, when the computer returns the word to the first position displayed. (I might note here that the program does not *always* find all possible locations for the entered word.) You then decide where you like it best, and place it there.

Before moving on to a new word, you have the opportunity to delete the most recent entry. Or, you may opt to enter the manual mode in order to change or delete previous entries. While in the manual mode, you are free to type over existing words, type in new words, delete words entirely, or even change the overall size of the grid.

When you enter a word that does not currently fit into the puzzle, the computer will store it in the unused word file. Should you later enter a word that does fit, and the unused word will connect to it, the program automatically positions *both* words (or as many as will now fit) into the puzzle grid. If you choose to ask the computer to check for alternate locations for the entry, it will relocate all words that were entered simultaneously, but not any words that were entered in previous turns.

Once you are satisfied with the word list, and the position of the words in the puzzle grid, you

can save it to your formatted data disk, where it will be stored until such time as you decide to play it, edit it, copy it to another disk, print out copies of it, or delete it from the data disk altogether. All the above options are available from the main menu screen. You can even save partially-worked puzzles to be called up and finished at a later date.

The print mode offers files configured to be compatible with 36 different printers. With that many from which to choose, you should have very little problem finding one that will work with yours.

You can print out the puzzle grid with the unused squares left blank (as in most commercially-produced Kriss Kross puzzles), or you can black them in, forming a square puzzle field. You can choose whether or not to print out your word list (list of "answers" to the puzzle). You may also choose whether or not to print out your list of clues. You can even choose to print out a list of unused words, until the puzzle is "done". (When the puzzle is ready to be played, and is saved, the unused word list's job is over, and it is erased automatically.) A command to pause between puzzle, answers, and clues can be entered, causing the printer to pause between segments, so you can adjust the printer paper in order to have each segment printed on a separate piece of paper.

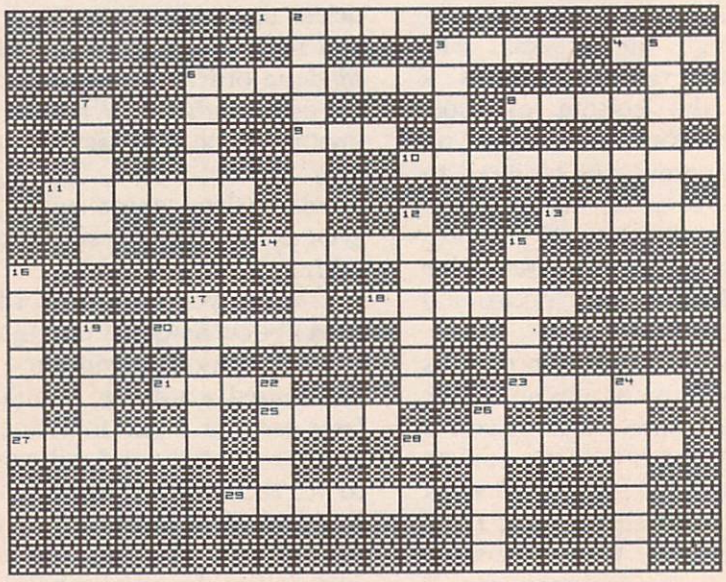
You may then play the puzzle, using only the grid and a word list, as Kriss Kross puzzles are worked. Or, you may choose to play the puzzle using the grid and the list of clues, in the manner of crosswords.

The edit mode offers ample opportunity to change existing answers or clues, as well as to add new ones.

The save mode is very user friendly. It is replete with prompts to help prevent you from erasing puzzles you wanted to keep. You can save your finished (worked) puzzle without erasing the unworked version. If two or more people wish to work the same puzzle, all versions can be saved without losing any others, by identifying them with the puzzler's initials.

Your puzzle can be saved at any point during the Enter Answers mode. You can also save your work from the Enter Clues mode, as well as from any point in the Edit mode. And, your puzzle can be saved at any point during the Play mode. In each case, prompts ask whether to replace the existing version or to save this under a new name (which, of course, you are then asked to provide.) You can save up to twenty puzzles on each formatted disk. The programmers get four stars for planning and execution here!

AT THE ZOO



Across

1. Primate of high intelligence.
3. Bovine animal used for labor.
4. Monkey.
6. Pachyderm.
8. Mountain lion.
9. "Wise" old bird.
10. Some call it a prairie dog.
11. Nature's lumberjack.
13. Slithery, skinny creature.
14. Eastertime keeps him busy.
18. Black and white and "red" in origin.
20. Jungle kings.
21. Smokey's cousin.
23. Well-balanced sea-farers.
25. Sharp eyes and a bird brain.
27. Andean beast of burden.
28. Boxer with a pocket.
29. Clown of the jungle canopy.

Down

2. Obesity is a way of life for this river dweller.
3. Be he sea-going or from fresh water, this furry one is fun to watch.
5. Pride is a trademark of this colorful animal.
6. Our national emblem proudly displays his splendor.
7. Ship of the desert.
12. Lots of dots with a bird's eye view.
15. Teddy bears.
16. An ape with a pretty nose.
17. Yipes! Big feline stripes!
19. Horse of the African plains.
22. Horny is one way to describe this big one's nose.
24. Spots (before *your* eyes, not his.
26. This animal can really bug you.

One minor blemish, if you can call it that, occurs in the Enter Clues mode. You can, at any time, return to Enter Answers to correct or change words. However, when you return to the Enter Clues mode, you will find that all of the clues previously entered are gone, and you must re-enter them.

You can choose the Correct Answers mode instead, from which you can type over *existing* letters in your answers, but you can't add letters in empty spaces. On the other hand, you do *not* lose your typed-in clues.

If, while you are entering clues, you see something that you wish to change, you would be far better off to make note of it, finish entering your clues and save them. Then, use the Edit mode to make the corrections desired. You can edit your clues from the Edit mode, as well.

Once you have entered and saved your clues, (and done any needed editing), your puzzle is ready to play. You can either play the puzzle on screen, or print it out, complete with numbered boxes for word beginnings, and blacked out squares where no letters are to go. However you choose to play it, **Crossword Magic** will help provide you with many hours of puzzle enjoyment.

One fun thing I ran across is to fill the unused word list with 30 or 40 words, and then enter them all at once! Instant crossword. Then push the right buttons and watch the computer relocate all of the words, making a new puzzle. Use your imagination. Now is your chance to get even with all those commercial puzzle writers ... make up words of your own!

Accompanying this article are two puzzles created with **Crossword Magic**. "At the Zoo", so named for obvious reasons, is offered with clues, to be played as a Crossword. The second puzzle, "O My!", is offered with a word list, to be played as a Kriss Kross. Granted, there are no serious brain-teasers in these puzzles, but I promise none of the words are homemade! Have fun.

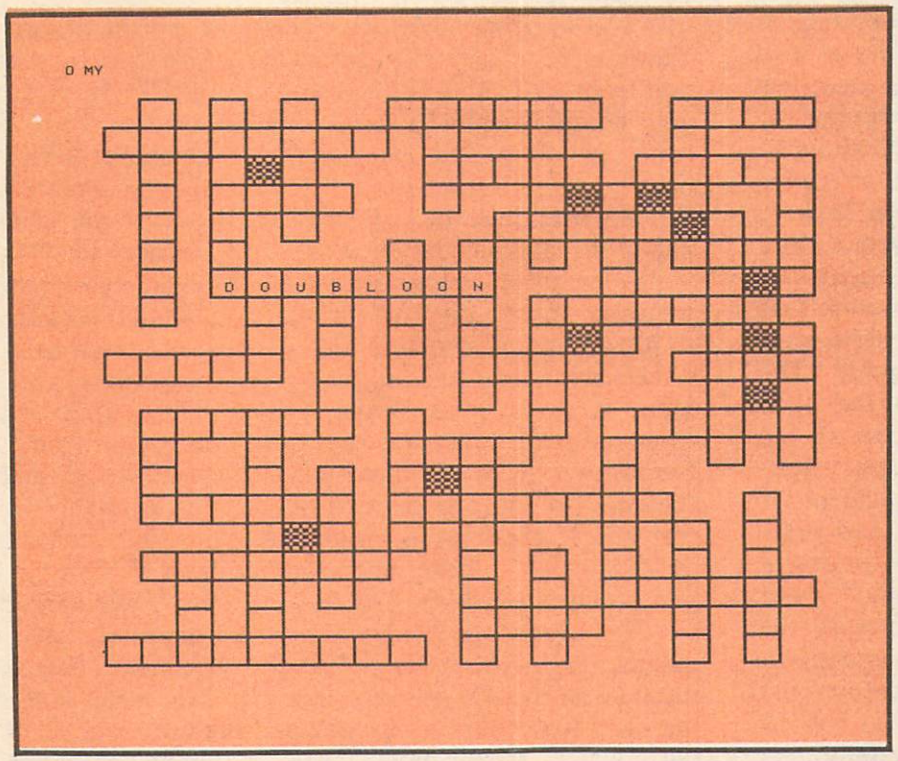
Crossword Magic is available at your local retailer — suggested retail price, $49.95. For further details, contact:

Mindscape, Inc.
3444 Dundee Road
Northbrook, Illinois 60062

Or, the distributor:

Gessler Educational Software
900 Broadway
New York, NY 10003
(212) 673-3113

■



Four-letter words

Boos
Logo
Loot
Moon
Mood
Oboe
Oleo
Onto
Oops
Oslo
Soon
Wood
Zoom

Five-letter words

Cocoa
Color
Flood
Honor
Icons
Loose
Onion
Ozone
Photo
Polio
Rodeo
Sloop

Six-letter words

Baboon
Baloon
Goober
Odious
Option
Orator
Poison
School
Smooth
Snooze

Seven-letter words

Dogwood
Obelisk
Octopus
Portion

Eight-letter words

Doubloon
Oncology
Oncoming
Operator
Overlook

Nine-letter words

Opportune
Possessor
Professor

Flexidraw Sets Standard For 64 Light Pens and Software

by Mindy Skelton and Kelly Allen

When you consider the graphic tools available in today's market for the C-64, Flexidraw 5.0, by Inkwell Systems is in a class by itself. The ease, speed, and flexibility of this system, in our opinions, easily put it on a par with the Macpaint drawing system of the Macintosh (and we might add that if you were to look at similar pictures produced on the two systems, you might conclude that the Mac is *slightly* overpriced). This system does so much (and does it so well) that we are not going to attempt to fully explain every aspect (Randy does want to print a *few* other things in this issue), but we will try to give you a complete enough overview to enable you to get a feeling for just how remarkable Flexidraw is.

One of the most attractive features of Flexidraw, at least to us, is how quickly we were up and drawing. Half of us thought that the system was clear enough that the manual could just serve as a quick-view reference guide. The other half of us thought that even though the way to access all the functions is not *completely* self-evident, the simple expedient of reading the commands, and practicing for a few hours, asking a few "How do I do this one?"s and throwing the manual across the room once or twice was enough to get things going. We (both of us) found the system a real joy to work with once we got going.

One of the few things which took some getting used to was the angle at which one has to draw when using a light pen. The kinesthetic desire to draw as one would with a pen or pencil was erased (or at least significantly lessened) by a few hours of use. The ideal situation would probably be to sink your monitor into a table top, but few of us are willing to go to this extreme.

The feel of the pen and its quickness and ease of response, led us to think of it as a real artist's tool rather than just a toy light pen. We were especially pleased by the reactivity of the top-mounted switch. Only a light application of pressure (no more than might be expected with any fine drawing instrument) is necessary to get you on your way to a real drawing adventure. This lightness of touch will also save you from scratching the screen of your monitor, and will, in the long run, save wear and tear on the pen.

We examined the inside of the pen (not recommended — it voids your warranty) and we must say we were impressed. The decision to use an optical switch instead of a mechanical one was, in our opinion, a great idea. What this means in plain English is that the Flexidraw pen is activated when the head of the pen retracts into the body, breaking a beam of light. Simple . . . beam unbroken, pen off; beam broken, pen on.

The advantage of this over a mechanical switch is that a mechanical switch employs moving parts to accomplish its task — parts which, as you might guess, wear with time. The Flexi pen, on the other hand, should give years of reliable service.

Extreme care has been taken in the design of this pen to provide adequate grounding, thus eliminating any possible shock hazzard. Even the pen's cable and joyport adaptor display the same "no compromise" approach to design that is noticeable in the rest of the package. The pen is fully serviceable and should perform long after the two-year warranty is up.

Once you load Flexidraw, you will notice an area at the right of your screen which shows what Flexidraw refers to as a "dynamic menu". This means that the menu options listed show only the items currently available, and can change as you go "into" a menu. For example, on your initial screen you have a choice of DRAW or SKETCH. If you touch your pen to the word SKETCH it will change to SPRAY while the rest of the menu remains the same, but if you touch DRAW it not only changes to POINT, but you are given new menu options: FILL, ZOOM, PASTE, and TEST.

Touch your pen to POINT once more, and it becomes RUBBER, while PASTE and TEST disappear from your screen. If you press the left arrow key (the one at the top right of your keyboard . . . not the cursor key) while the option POINT is showing, a small cross will appear where the arrow that marks your cursor position is pointing. Not only that, but three new menu options appear: LINE, BOX, and CURVE. Now if you move your cursor, and then touch the pen to one of these options, you get an automatic line, box, or, with CURVE, three more options:

ARC, ELIPSE or CIRCLE. Get the idea?

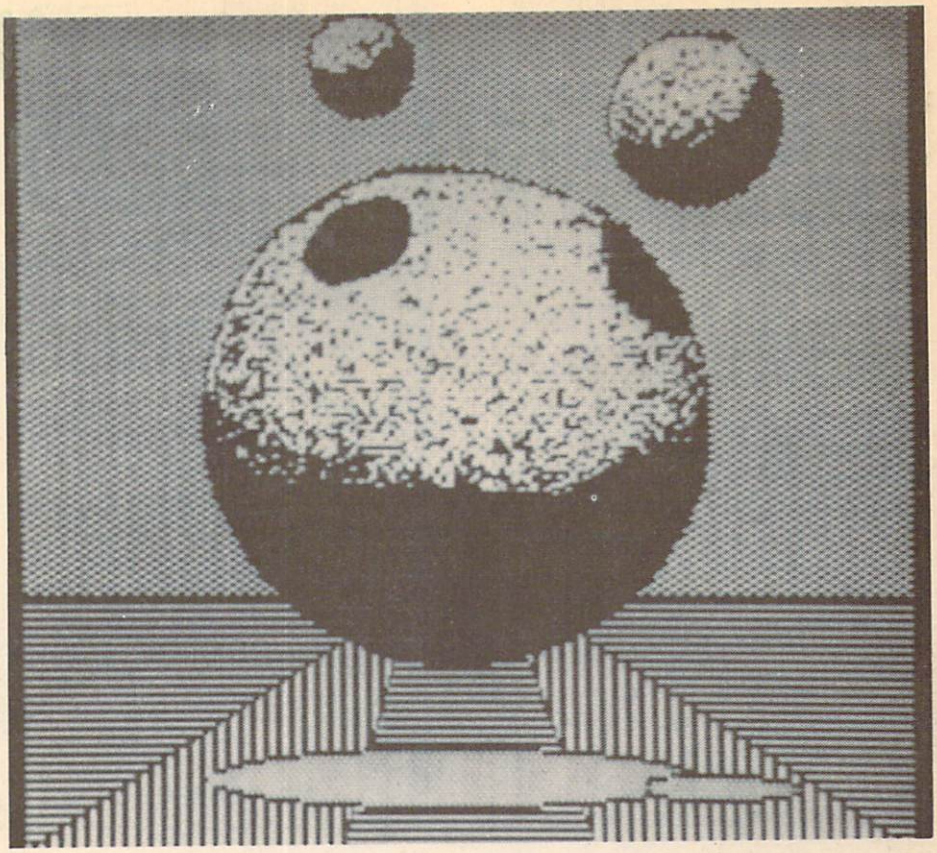
Now that you know sorta how it's done (touch the pen to the screen to select the option you want, draw with the pen to make your images), let's talk about just what the capabilities and limitations of this system really are.

As you may already have gathered you can do freehand drawing, but let us add that you can do this either with a smooth line or with a spray-paint effect, in one of three different line thicknesses. You can generate automatic boxes, arcs, elipses, or circles, again in different line thicknesses.

You can also generate lines in two different ways; point-to-point lines where you select the two end points and the system provides you with a connecting line, or rubber banded lines where you select the starting point and stretch the line around the page until you find the place you want it. Nice, but hardly enough to cause someone to lay out the rather hefty price of this system (the list price of Flexidraw is $149.95). What else can it do?

Well, of course, you can invert the screen. This makes it simple to "erase" small sections by inverting and drawing in the suddenly light areas, thus causing them to "disappear" when the normal screen returns. Looking at the inverted image also gives you a new perspective which is always helpful when you are creating an image.

If you don't want to clear the entire screen by the "inverse" method, you can also clear the work area by simply pressing the shifted CLEAR/HOME key. You also have a choice of two separate work areas, called pages. You select the page you want by touching your pen to PAGE. Still not enough to make you shell out the big bucks?



You can FILL specified areas by selecting one of the patterns offered from the FILL menu, or by loading in a custom font called PATTERN (we'll talk more about fonts, and loading and saving files later) which gives you 14 additional patterns (many of which will be quite familiar to you if you've played with Macpaint), or even by selecting an alphanumeric character from one of your fonts (more on this later) to use as a FILL.

Flexidraw also provides an interesting option called SHADE which allows you to either overlay fill patters on dark areas of your screen to create new effects, or to remove all but the outline of a drawing or outline an edge.

Another Flexidraw option lets you ZOOM in on any area of your image to work on a pixel-by-pixel level (every pixel is magnified to "character block" size), in either normal or inverse screen for highly detailed work. [Note: Character blocks are the 8 pixel by 8 pixel blocks which make up the screen and are so familiar to those of you who have played around with color on the C-64.]

While in ZOOM mode you are working in a "window" which magnifies a section of your image. A reference window shows you the magnified area in its normal size so you can see the effects of your work as it happens. You can move your window around in the zoom mode by touching your pen to the arrows at the bottom of your screen, and you can move your "reference window" from one side of the screen to the other by simply touching it with your pen.

Yet another option is to GRID your screen. The GRID is a normally invisible crosshatching of lines formed by the borders of the character blocks (which can be made visible by pressing *f1*). While GRID is in effect, all shapes and text are placed along the lines bordering the character blocks. The placement of images along these guides makes lining up text on your screen and matching

shapes from one screen to another easier, but it makes freehand curves impossible.

In addition to GRID, you can select CROSSHAIRS, two movable lines, the intersection of which is moved around the screen by your pen. CROSSHAIRS, like GRID helps in matching the alignment of shapes and text, as well as guiding the positioning in COPY/PASTE operations.

Now that we've mentioned it, let's talk a bit more about COPY/PASTE. Flexidraw lets you copy and paste any image between screens by creating a copy of your image on a "scratch pad". The section of the screen you wish to copy is captured while the GRID/POINT options are in effect.

By touching the pen to the screen and using the COPY option, a box is formed and all the image in the box is copied to the scratch pad without removing the

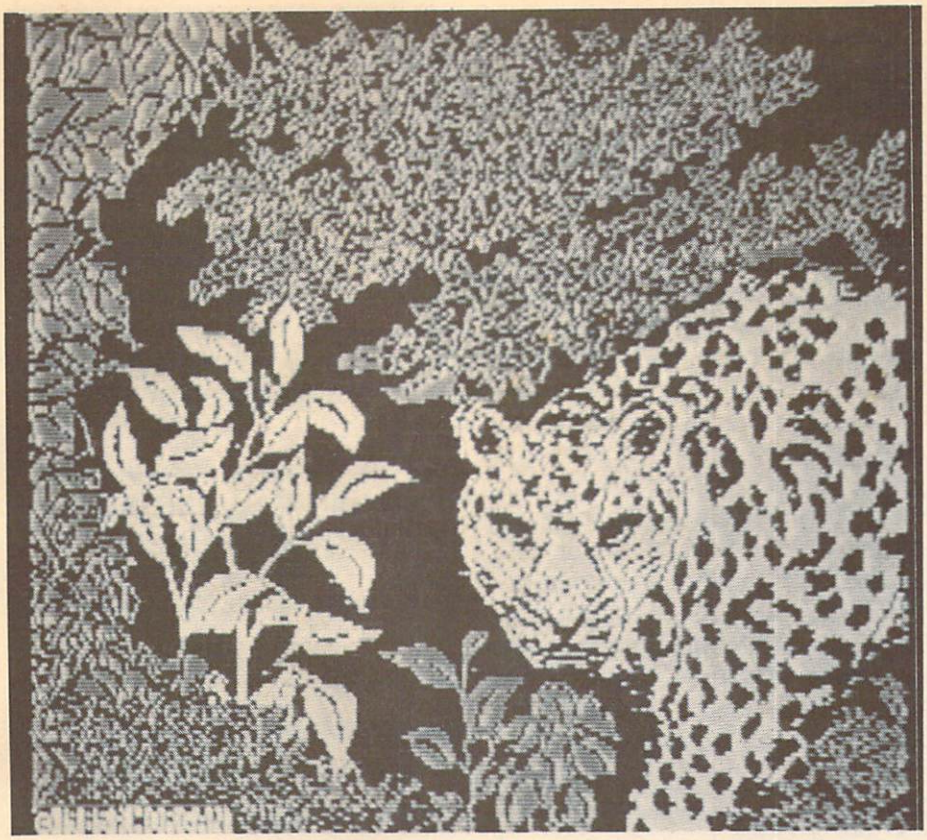


image from your original screen. The image on the scratch pad can then be flipped vertically or horizontally, or rotated (in 90 degree increments). Once the image is the way you want it, you can place it on either of your screens in one of several ways.

You can position the image in either GRID or PIXEL mode by using the pen or the cursor keys. Even then, you have more choices. You can try out a particular placement with the PASTE/TEST command. If you don't like how it looks, just try another PASTE/TEST and your previous placement will be undone. You can even make permanent transparent or opaque overlays with PASTE/WITH or PASTE/OVER. If you decide you don't like your permanent position, pressing RUN/STOP *BEFORE you do anything else* will "undo" your placement.

I hope you're not getting tired of hearing all that Flexidraw can do, 'cause there is a *lot* more. If you need to go out for a cup of coffee now, I will understand.

Back? OK. In addition to the shapes you can create, the program offers you the capability of adding text and numbers to your screens. These alphanumerics can be treated as any other shape, in that they can be copied, pasted, flipped, etc. There are built-in fonts ranging from standard Commodore upper case/lower case/graphics, to Gothic, Art Deco and Computer, with several other stops on the way. Text is positioned and edited with your cursor keys, and can be expanded as needed. Flexifont (a companion package) offers even more fonts (33 to be exact) and the capability to generate other fonts.

Whew! After all that work, you don't want to just shut off your system and lose that brilliant piece of art, do you? Well, you don't have to. Flexidraw gives you the option to print your art as either a partial- or a full-page image. (A chart is supplied to give you an idea of where on the page your particular printer will place the image.)

A number of printers are supported, and helpful hints are supplied as to how to get the best from your particular printer/interface combination. If your image took more than one screen, you can print both pages on a single sheet of paper using an option which allows side-by-side printing.

If you want to be able to save your picture now for future use or editing, you can save it to your disk (not the program disk, of course) by entering Flexidraw Filer and choosing from the Filer menu. This is the option you choose to load files, check your directory, format a disk, etc. Never say Flexidraw doesn't give you options.

Two additional options are available, if you are less artistically inclined than you might like. One, Flexidraw gives you libraries of pictures and shapes which you can edit, combine, color and generally experiment with to create new images. Everything from electrical schematics to the Brooklyn Bridge is there for you to use. If there is more that you need, there is another companion program called Graphics Galleria available.

If this is *still* too artistic for you, how about just taking a photograph and coloring or altering it? Fine! — Flexidraw interfaces with both Computer Eyes and the Micron Eye Digitizing System. You can have great fun capturing images from life, or a VCR, and transforming them via Flexidraw into something new and yours.

OK. So now you have the picture drawn to perfection, but it's all in black and white. Is that the way you want the final form? If it's not, there's a way to add all the color you could want. An option called Pen Palette allows you to select the foreground and background of your color blocks (8×8 pixel blocks). You select a

color by touching it with your pen, then you move the pen to the part of the picture where you want to put it. If you change the color in a particular "pot", all instances of that color on the screen will change to the new color. You can also add color on a pixel-by-pixel level for fine detail work by entering the EDIT mode.

So, now the picture is just the way you wanted. How can you record this for posterity? You can SAVE it to disk from the Pen Palette Filer (along with the color information for the image), or if you have a color printer, you can print out a colored image.

This is all pretty impressive, but, it's not all. Flexidraw also includes a quick, easy and wonderfully useful sprite editor, and animator. There is a program called Transgraph which allows you to transmit your Flexidraw images via modem to other Commodore computers.

For those of you a bit more adept in programming techniques, there are routines included to use Flexidraw pictures in your BASIC programs, and even to access Hi-Res graphics routines which will quickly clear your screen, draw circles and lines, plot points, etc. Any of you out there who have ever tried to do bit-mapped graphics on your own will really appreciate this feature.

We're afraid we've really just skimmed the surface of this program, and not done it justice, but we've at least mentioned the basics of Flexidraw. No amount of text can take the place of hands-on experience, so if high-quality, professional-looking graphics are important to you, if you want the best graphics system available for the Commodore, and if you are willing to spend a few dollars, spend some time with Flexidraw.

Command Line Interface: Your Direct Line to AmigaDOS

by Grant Johnson

The Amiga has two main ways of interacting with the user; the graphics-oriented approach of the Workbench (which we looked at last month), and the Command Line Interface. The commands in the latter case are issued primarily from the keyboard.

The point-and-shoot control of the mouse is quick and easy because you are allowed to "gloss" over the details. That is as it should be, since mouse systems requiring too much attention to detail leave you flailing about like a conductor in mid-crescendo. There are just too many nooks and crannies to visit to get anything done, but with ten fingers and ninety-some keys on the job, keyboard control, by its very nature, is suited to detail.

Blessed with the best of both worlds, the Amiga offers a choice. If you are uninterested in, (or cannot cope with), the details, you can shoot your commands from the hip through Amiga's Workbench. Most of the time this suits me fine, but when I need (or want) to get closer to the Amiga's internal realities, I go for the Command Line Interface (CLI).

Good news, bad news

Unfortunately, the manuals that come with the Amiga (*Introduction to Amiga*) only mention in passing that there is an alternative. The information needed is hard to find — not because it was never published (the full set of Amiga manuals total over 4000 pages and may be the *best* ever written for a new machine), but, because Commodore chose not to be your publisher. They instead sold the rights to third parties.

The DOS (Disk Operating System) information was scheduled to be available by this time from Bantam (under the title *The AmigaDOS Manual*, and much of the remaining literature is coming from Addison Wesley. In the meantime, the machines are in good supply, but manuals are not.

Fortunately, most of what you need to know is right there in the machine. All you need is a little help to get at it. Most of us were taught BASIC by a computer, and there is no reason the Amiga can't teach us AmigaDOS. A little orientation at the outset will save you a lot of grief, and it is my intent to provide that orientation here. By the end of this article, you should be in a position to learn for yourself what you need to know through the time-honored method of "try it". Now to the orientation ...

Processing

According to Amiga literature, every job assignment that the machine is given is to be viewed as (ahem) a "process." Thus, you have file printing processes, disk handling processes, and word processing processes. (Well, what are microprocessors for, after all?)

Actually, far from trying to sound "sophisticated," this approach makes practical sense once you become familiar with the Amiga's abilities. The chief distinction between programs, sub-programs, sub-routines and the like is how they are begun and ended.

Programs terminate by giving control back to the operating system, and sub-routines give control back to the program that called them. But, in a multi-tasking machine, everything is grist for the mill. You may have programs working along side of other programs while they share information and resources. (What is a sub-routine submissive to when it may be used by several programs at once?) And virtually *everything* that happens is subject to priorities, scheduling and interrupts. It makes sense, then, to think of all such entities as processes, whose place in the scheme of things will be defined by circumstance.

Natural

Dealing with processes is a far more natural way of doing things than having to observe the niceties of organizational hierarchy. You can think in terms of doing this and then doing that — or even doing *several*. The only reason you have had to deal with distinctions and sub-distinctions in the past is that you had to do for the machine what it could not do for itself.

The Amiga is a different kind of partner. It works mightily behind the scenes to coordinate and assemble the elements you require. It even copes with conflicts, and only rarely admits defeat.

Still, you *do* have to tell it what you want it to do. Enter the Command Line Interface.

It has been observed that the single most important quality associated with human intelligence (at least the kind measured by I.Q. tests) is long-term memory. When you know (remember) how something is done, then you're a smart fellow, right?

The corresponding elements in today's computers are storage devices such as disk drives; and the first thing an Amiga does after it wakes up is to reach into disk storage. It drinks deeply from this magnetic well of knowledge as it marshals its resources, from the hardware up.

Starting CLI

If you follow normal startup procedures, the Amiga begins by offering you the Workbench environment. Open the Workbench on a factory-fresh computer, and you will look in vain for any indication of how to tell the machine you want to enter text commands.

Obtaining the CLI option is accomplished by opening the "Preferences" window. On the resultant preferences screen, you will find an "on/off" gadget next to the letters "CLI." Turn this on, save the preferences, and the next time you start Workbench you will see a cube-shaped logo with the letters CLI under it. Select it, and a window will open with a `"1> "` and a waiting cursor.

The sharp-eyed might be puzzled by the caption on this window, "`New CLI Window.`" New? Where was the *old* one? During the Amiga's startup sequence, a screen briefly appears entitled "AmigaDOS." Contrary to first impressions, it is AmigaDOS that starts Workbench and CLI that does the loading (with a special command `"LoadWB"`). DOS looks for and executes a series of CLI commands located in a disk file called "Startup-Sequence". It is this sequence of commands that prints "Workbench disk. Release 1.1", etc., on the screen.

If you interrupt this sequence by holding down the CTRL (control) key and pressing "D", the Amiga will respond with `"**BREAK - CLI"`, and present you with the `"1> "` prompt.

About that "`CTRL D`" stuff: CLI supports a system of "attention flags". They are `CTRL C`, `CTRL D`, `CTRL E` and `CTRL F`. They can be used to effect the progress of on-going processes. `CTRL C`, for example, will usually cause a command to be abandoned (shades of the RUN/STOP key on a C-64 or a 128PC. `CTRL D` might cause the command to go on to the next task in a series. Generally, the further down the alphabet you go, the less drastic the effect. But, the exact outcome depends on immediate circumstances. This is *not* a "hardware" event, a flag is set which may or may not be checked and acted upon by a particular process — you have to know the process.

It all makes better sense after you've played with it a while. But, for now, it is enough to follow one of these methods of opening a CLI window as a recipe.

What to Say After 1>

If all goes well, you should see the `"1>"` prompt (the word READY serve much the same function in the '64). This may look a lot like the `"A>"` seen on other machines, but there is a big difference. The `"A"` in the other systems tells you that disk drive `"A"` is the current drive. The `"1"` in the AmigaDOS prompt has nothing to do with disk drives. The number indicates *which CLI process* you are looking at (there may be many).

Let's start with an example. With a blank diskette in the internal drive, we'll format (or "NEW") it for use.

```
1> FORMAT DRIVE DFØ: NAME "E
xample"
```

The Amiga responds with "Insert disk to be formatted and press RETURN." After we do so, the Amiga first formats, then verifies the disk, printing a running total of the cylinders or tracks as it goes. If it encounters any trouble along the way, it will tell you about that, too.

The command here is FORMAT. When the Command Line Interface finds that word, it loads the code that performs this process from the system disk. As a matter of fact, it comes from within a directory called "c" (for command) and its file name is actually "Format".

The Workbench disk cantains the system information, and is the one most people use at first. If you had removed the system disk before issuing the command, the Amiga will flash a message to you to replace it. It is safe to do so, because the machine will go on with the "Insert disk ... " message before anything nasty happens. (You should, in any case, backup your Workbench disk — and, maybe, keep a copy with a trusted friend in another county.)

DRIVE and NAME are called key words and, in this case, are required. DFØ: is the device name of the Amiga's internal drive (there can be up to four drives — DFØ: through DF3:). Note that the ":" is a part of the name.

DFØ: is analogous to Device 8 on a 64 or a 128. The string in quotes is the name given to the disk itself, and is the name you will see below the disk logo in the Workbench. The quotes are optional this time, since the name contains no spaces.

If you get lost, the Amiga can give you several kinds of help. If I had made the error of leaving out the key word NAME, the following would have been seen on the screen:

```
1> FORMAT DRIVE DFØ: Example
Usage: FORMAT DRIVE {DFØ: | D
F1: | DF2: | DF3:} NAME "MYDIS
K"
```

The machine is offering a model (Amiga calls it a format) of what it expects. Items enclosed in braces and separated by vertical bars read, in "English", choose DF0: or DF1: ... Yet another way of getting help is to follow the command in question with a question mark.

```
1> FORMAT ?
DRIVE/A/K,NAME/A/K:
```

Amiga calls this a template, and we should take a minute to clarify the symbols used. "/A" means that an argument is required. In the above, you must specify a drive (DF0:, DF1:, etc.) and a name, even if it is " ". The "/K" means that the keyword must be used; FORMAT without DRIVE and NAME will not do.

While I am thinking about it, capitalization *is* optional. Format is the same as fORMAT. There is one other symbol "/S" that indicates that a keyword is a switch and needs no argument. Notice that the command name itself does not appear in the template. You wouldn't have gotten far enough to see the template without it. The final ":" is a separator and the cursor waits to its right for your next attempt.

There is one last form of help that the Amiga will give you. When something goes wrong and you get an error message, you can always ask WHY.

```
1> JOIN
Bad Args
1> WHY
Last command failed
because argument line
invalid or too long.
```

I entered the command with none of the required arguments. The computers response to WHY can sometimes be quite helpful — particularly when you think that "args" are the noises that cartoon people make when they are strangled.

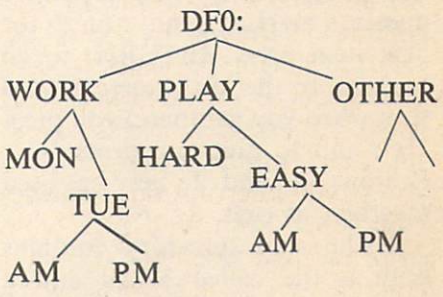
Now that you know how to open a CLI and how to get a some help from the machine, you need to know something about where on the system disk to find a list of commands for further exploration. And for that you need to know about ...

Disk Organization

Locating the things you need in disk storage is relatively easy once you understand its organization. To begin with, there might be more than one disk device. As mentioned earlier, the drive that comes mounted within the body of the Amiga is designated as "DF0:" — Disk File Zero. DF0: is the starting point from which you may reach any file on that drive. The colon is a convention used to emphasize its special "device" status. (There are other devices besides disk files. PRT: is the printer file, CON: is the name of the console device, RAM: is the name used for the random access memory device and there is even a NIL: device to send things to never-never land.)

This starting point is sometimes called the "root". The name comes from the logical structure used, a "tree" structure. A tree structure is enormously flexible. It can be as simple or as complex as you need. Formatting a disk creates a root directory, and if all you did from that point on was to store things on the disk, you would have a structure recognizeable to anyone with a 1541 — a disk directory with a series of files in it.

In the Amiga, however, any of these files could also be directories with entries of their own.



In this diagram we have the root directory (mounted on DF0:) which contains three directories

(WORK, PLAY and OTHER). The command to use to see a list of directories is:

```
1> DIR
    WORK (dir)
    PLAY (dir)
    OTHER (dir)
```

Now on a 1541 disk, one directory is plenty, but, when you consider that an Amiga disk contains 880K of room and that this is probably the smallest unit of storage that will ever be used with this machine, you can see the need for a division into smaller units. Just finding the item you want in a list of, say, a hundred file names would be a pain — to say nothing of trying make sure you were using a unique name each time you stored something.

Going a step further with the DIR command:

```
1> DIR OPT A
   WORK (dir)
      TUE (dir)
       AM      PM
    MON
   PLAY (dir)
      EASY (dir)
       AM      PM
    HARD
   OTHER (dir)
```

The OPTion A added to the DIR command tells the machine that I want to see all of the files on the disk. There is another option, OPT I, that allows you to work your way through the disk interactively. Then the directory and file names appear one by one with a question mark. To move on to the next item press RETURN; to go back up to the last directory (and stop when you get there) you press "B"; and to quit you press "Q". Options A and I can be used together, as well.

The root directory contains both a file called MON and a directory called WORK which, in turn, contains two files, AM and PM. Note how the file structure is represented through the use of indentation (multiple files are display in two columns). The root directory also contains the directories PLAY and OTHER. As you can see, PLAY has a directory, EASY, and a file, HARD, in it and the EASY directory contains files AM and PM.

One of the more useful commands is TYPE. It reads a file and types its contents (to the screen, if you don't tell it otherwise). All the files in the Example disk happen to be small text files. We can use the TYPE command to illustrate a bit more about disk structure.

```
1> TYPE DF0:WORK/MON
This is mondays work.
```

What we are dealing with here is something called a path. TYPE is the command and is followed by the identifier of the thing we wish typed — in this case, "DF0:WORK/MON". If you look back at the diagram of our example disk, you can see the path leading to the item we wish. It starts with the device name DF0:, moves on to the directory WORK and finally arrives at the destination, MON.

```
1> TYPE WORK/TUE/AM
This work was done before
noon.
1> TYPE PLAY/EASY/AM
Took the morning off for
easy play.
```

Note that even though there are two files with exactly the same names, there is no confusion since they have different paths (and directories). Note also that I omitted the device name DF0:. I could do this because the root directory (device name, in this case) is the "current directory." With the Amiga, you have the freedom to make any directory the "current" one. This is done with the CD (current directory) command.

```
1> CD WORK
1> DIR
   TUE (dir)
  MON
```

CD uses the same path conventions as other commands.

```
1> CD :PLAY/EASY
1> DIR
   AM          PM
```

When you wish to explore another branch of the file structure, you will need to begin your path back at the root. If you are not changing disk devices, the colon by itself is an adequate stand-in. On the practical level, what may look like a lot of work just to keep things separated turns out to be quite convenient. Tuesday morning I might open move the directory to WORK/TUE/AM, and spend my whole morning without worry about anything else on the disk.

Now that I've gotten you into all this, how do you get out? To close a CLI window, you type ENDCLI. If you entered the CLI from Workbench, you will find

yourself back there. If you started with CTRL D, then to get to Workbench you will need one last command.

```
1> LOADWB
1> ENDCLI
```

After the first of these commands loads Workbench, you will need to select the CLI window with the mouse before you can close it with ENDCLI.

Graduation

Now we can put what we've covered to some real use — seeing the menu of commands at our disposal.

```
1> DIR C

Assign          Break
CD              Copy
Date            Delete
Dir             DiskCopy
Echo            Ed
Edit            Else
EndCLI          EndIf
Execute         FailAt
Fault           FileNote
Format          If
Info            Install
Join            Lab
List            LoadWb
MakeDir         NewCLI
Prompt          Protect
Quit            Relabel
Rename          Run
Say             Search
Skip            Sort
Stack           Status
Type            Wait
Why
```

There will be a flood of command names scrolling through your CLI window. If you are not a speed reader, you will want to suspend this rush of good stuff. The screen is the place where the Amiga and you meet to exchange information — you both write on it. The Amiga knows who's boss, however, so if you indicate through the touch of even a single key that you wish to type something on the tube, it stops immediately. As the command list fills the screen, just touch the space bar and the listing will freeze. To continue, you can enter a dummy command by pressing RETURN, or you can take back what you might have been about to say with the BACK SPACE key.

Next month we will look at how to deal with multiple CLI windows, transferring files and data streams between devices and much more. To tempt you back, here is a command to use so that you can get a printed copy of the CLI commands.

```
1> DIR > PRT: C
```

■

Computer Widow's Compendium

by Lyn Chase

The other day I was talking with some staff members and made an interesting discovery. Lots of computer junkies like to leave their computers on ALL THE TIME. Some of them say that it's better for the computer. Some say that it makes them feel like a friend is nearby. Some even say that it gives them a sense of security. And then, of course, there are the ones who run bulletin boards. There are even computer junkies who don't like to turn the thing off because they fear that it will trigger a depressive episode in their Little Computer People.

Since I learned the other day that my husband is not the only one to leave the beige brute ticking ad infinitum, I have begun to wonder if, perhaps, computer junkie-dom comes on insidiously — first, a slight nagging to know that the thing still works, then a desire for the security of knowing that one can call another "home computerist" on the modem without wasting that interminable thirty-second "power-up" time.

And, so it goes. Let's look at some of these reasons for boosting the power company's revenues so that we may be better able to understand our spouses' addiction.

Word has it that the computer and/or it's various and sundry components (power supply packs in particular) last longer when they are not subjected to frequent changes in status. There is no scientific evidence that has been brought to my attention that will either confirm or refute this theory, but the office staff tells me that their personal "personal computers" have been trouble-free when left on for months at a time. The question of the day is this: Would they be trouble-free if left OFF for months at a time?

One staff member tells me that she has a heightened sense of security when she can see the cursor flashing any time she looks in the direction of the computer. She compares this feeling to her children's desire for reassurance that the batteries in their toys still function — and the tearful pleas to "fix it" when the batteries fail.

I suppose that I can understand this. I feel that wonderful sense of security every time I get in my car and it starts when I turn the key. But even with the price of gas dropping, I still don't have the desire to leave my car running while I sleep . . .

There are some people who see the computer as a friend. After all, it can help you with your work (assuming, of course, that your work does not involve manual labor), it can entertain you, it can visit with you, it can even psychoanalyze you. Some hackers say that turning the computer off is like sending their best friend home. Aw, come on, guys! Can your computer down a six pack with you? Can it slip into something sexy and mix you a gin and tonic? Can it commiserate with you over the break-up of the Beatles? If you must have your computer, then have your computer. But if you want a friend, talk to your wife.

We have yet another category of "leave-it-on-ers". These are the ones who run bulletin boards. A computer bulletin board is a computer that can be called by other computers to leave messages, read

messages, "chat" with others of a similar bent, even raise the consciousness of the home computer community on a particular topic. Some churches have computer bulletin boards to spread the word of the Lord. Other organizations use them to instruct callers on how to build an atomic bomb.

There's a veritable wealth of information on these bulletin boards, and they all have one thing in common. The computer must be turned on and the appropriate software loaded for callers to be able to utilize them. So lots of people have an extra computer and an extra telephone line just for their bulletin boards. And, of course, they leave them on all the time. What do you suppose will happen if we have another energy crisis?

One last category of "leave-it-on-ers" is a newly-emerging group of "computer enthusiasts" who have discovered that there are Little Computer People living in their computers. (Actually, this is a game by Activision.) Those in this category who are developing relationships with their Little Computer People (LCP's) leave the computer on all the time in order to participate in the LCP research project.

LCP's must be fed, watered and petted. If they are neglected, they get sad. Then they get sick. I don't know what happens after that. If their needs are met, they will play music for you and will play games with you. The rest of the time, they go about the business of living. They exercise, feed the dog, read the paper, even have telephone conversations, presumably with other LCP's. They even go potty and brush their teeth.

Researchers note their activities in order to learn more about this newly-developed species of earthling. Needless to say, LCP's don't like to have the computer turned off. And researchers cannot do research when they cannot see their LCP's. So the computer stays on. I hear that some researchers even have an extra computer devoted entirely to the observation of LCP's. I've got a husband, a child, two dogs, two cats and a bunch of fish to take care of. Randy, take note: any LCP's that pop up in this house are your responsibility.

I nearly forgot one last category. There are some hackers who do not eat, do not sleep, do not go potty. They simply sit in fromt of the computer and compute. These zombies do not turn off the computer. If you have one of these in your household, you have my sympathy.

As you can see, there are several categories of "leave-it-on-ers" and if you are reading this, it is likely that there is at least one in your household. Do not despair. It's not contagious.

■

Nine Princes In Amber:
Fantasy, Sorcery and Intrigue

by Mindy Skelton

You're lying in bed ... you're only semi-conscious, but you notice you have casts on your body even though you feel no pain. A man is coming toward you with a hypodermic needle in his hand ... you must stop him from injecting you again, or this time you won't wake up. You *must* escape.

You're free now, but your memory is gone ... you don't even know who you are ... but you know you must get back to Amber. Amber — the one perfect world of which all others (including this one in which we live, gentle reader) are only imperfect copies. You quickly discover that you are Corwin, prince of Amber, you are unusually strong (the word superhuman springs to mind). You have quite a number of siblings, not all of whom seem to have great love for you (to say the least), and you need help in getting back to Amber and the throne.

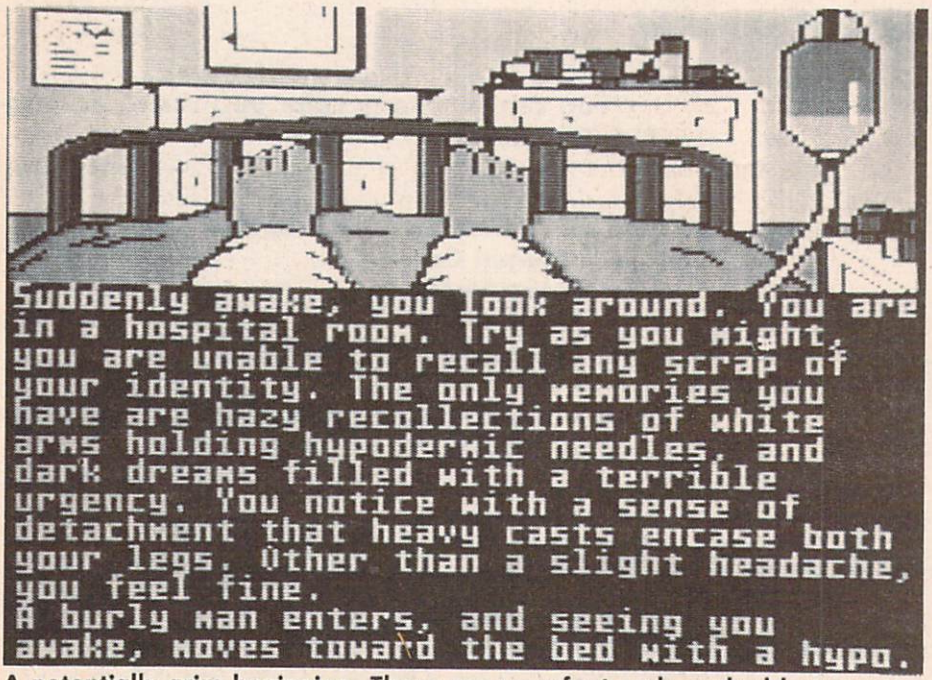
With this slightly breathless beginning, Telarium software plunges you into the gripping world of Roger Zelazney's *Chronicles of Amber*; a world of swords, sorcery, treachery and treason. With their usual superlative grace, Telarium has translated Zelazney's world into what ranks as my current favorite adventure game (sorry, Infocom). In fact, Telarium is doing such a nice job of translating various works to the screen (vid. *Rendezvous With Rama*, *Farenheit 451*) that they could easily become the successor to Classics Illustrated Comics.

The demise of that painless way of swallowing literature has left a void in the current generation of children's literary background (come on - did you *really* read *Last of the Mohicans* ??). Who knows, if this sort of thing catches on (in forms with a few more words than the Windham Hill Classics), reading may come back into fashion. (Well, I can dream, can't I?) Oh ... excuse me ... back to Amber.

To capture the full scope of the **Nine Princes in Amber** is no small feat. In this rendition, two double-sided disks are needed. Although this leads to more disk swapping than is usual in a text game, it seems reasonable when you consider the number of dimensions the disks must hold. Oh, yes ... in addition to text, Telarium provides some very nicely-done (if somewhat horizontally-distorted) graphics which are presented via split screen techniques (both horizontal and vertical splits), which enhance rather than detract from the text (when you consider that this statement comes from a woman who has always disliked graphic adventures ... well ...).

If you prefer your games with no graphics, or if you've played them through several times and just don't feel you need them, the graphics can be turned off. Yes ... I did say "Playing through again." This is not a one-shot adventure. The possible variations are myriad. Telarium says there are over 40 possible endings (some more "correct" than others), and over 40,000 game variations.

As in real life, small things can set you off on a new path and lead to big changes. Your life may depend on whether you smile or shrug, question or agree, not to mention the havoc that can be



A potentially grim beginning. Those are your feet and you had better get them out of there before the guy with the hypo ...

wrought by trusting the wrong person.

As if the story line were not enough to make this a worthwhile game (OK ... so I'm a fan of Zelazny), Telarium has done a trend-setting (I hope) job of programming. Not only is there an excellent parser which understands complex commands, but the language of the universe of Amber, as presented here, (a language known as Thari) allows you to pack masses of information into a single word. Words such as "Agree" could mean "Agree with Flora that you will not do anything to harm her." Or "Shrug" could mean "Well, gee, Random, I don't know. Why shouldn't we attack Eric?"

Anyhow, it saves you a lot of typing and speeds the progress of the game. The game provides you with a lot of description and interest for your little inputs. To help you in your attempts at communication, you are provided with a vocabulary listing of Communication Verbs, (further divided into "Hostile", "Neutral", "Friendly", and "Response") and Action Verbs.

There are times, however, when words are not enough. You and your family are rather hot-tempered, and some of your siblings would rather kill you than chat. As a result, you are, from time to time, forced into swordplay. In most adventure games, if you were to engage in swordplay, you would simply type in "Attack (whatever) with sword" — or, even more simply, "Attack" — over and over again, until one or the other of you was dead. Not here. In Amber, there is a specialized vocabulary for fighting ... you thrust, parry, feint, cut, jump, go high or low. Your strategy is just as important as in a real contest. If you simply attack with no thought, you'll probably end up dead. No extra charge — just one more example of Telarium's lovely program.

The graphics included in Nine Princes in Amber are good enough to enhance. Text scrolls along side (or below) split screen display.

Speaking of lovely programs, did I mention the music? Telarium has included some music that is very, very well done. There are, in fact, what might in other places be called motifs. There is the "Amber" theme that starts the game and recurs from time to time. There are themes for particular individuals (I especially like Flora's music), and there is incidental music ("A little traveling music, professor.")

In order to regain your memory and, eventually, the throne, you must reach the fabled undersea city of Rebma (don't worry, you'll find you can breathe underwater ... *you're* of the royal house of Amber ... you can do amazing things), and "walk the Pattern".

Walking the Pattern is a special part of the game. You are presented with a graphic screen, blank except for five stars. You are given a choice of "pieces" which you attach to the path you walk. By choosing the proper path pieces, you may walk a path that connects all five stars, but never crosses itself.

At each veil (star) you regain more of your memory. (If you are still confused, rest assured that the information that comes with your game will make all clear. The documentation is supurb.) Much information lies hidden in the Pattern, but also much danger. It would almost certainly be in your best interest to SAVE your game before you enter the Pattern.

Oh, yes. You can save games quite easily. Up to 10 game positions can be saved on your prepared Save disk. This gives you the potential for returning to preselected places and try different gambits to gain different ends.

Now, please don't imagine that walking the Pattern is the goal of the game, even though it *is* necessary. The ultimate goal is Amber and the throne. Everything I've mentioned so far is contained on the first disk. There is another entire disk of danger, adventure and intrigue before you sit quietly (or not so quietly) on the throne.

In case you didn't notice, I like this game a *lot*. It gets a straight A on the Skelton Scorecard. Go get it. Play it. Get involved. See you in Amber.

■

Real Gamers Don't Read Instructions

by Robert J. Sodaro

Did you ever have one of those days? Yeah, me too, only it seems that mine has been happening for about a week or so now. Since last we chatted I've been quite busy looking over all the new releases and attempting to decide what to review. (I've also been busy doing lots of other stuff, but that's not important right now.)

I had originally settled on five products to discuss, but seeing as how I'm a devout Murphyist, the inevitable had to occur (You know Murphy — he's the guy who told us that if anything works, it's by accident ... I think he invented computers, by the way). Needless to say, I had to drop one game due to time considerations (*i.e.*, I goofed off to long to give it a proper looksee), and another disk decided it wanted nothing to do with me, my column, or the world at large, and refused to re-boot after the first play. All of this leaving me scrambling to make some last minute substitutions (and making me a liar in front of Randy). Still, all's well that ends well, as I think I made some viable choices. But you're not here to listen to my troubles, you wanna know 'bout the new games I've been playin'. So without further ado here we go ...

Our first pair are a couple for the kiddies from Fisher-Price — **Peter Rabbit Reading** and **The First Men in the Moon**. The first is designed for children ages 3–6 with the second being for kids from 9–12. The instructions (yeah, instructions, yuck) for both games bear a brief note to parents from Fisher-Price explaining the play of the game and the particular skills that they expect to teach.

Peter Rabbit attempts to build a child's grasp of phonetics by using a simple skill menu. Many of the lessons use a voice output to challenge the child to identify missing letters based on the sounds they make. **First Men** combines numeric and word problems to teach five levels of math problems.

To digress for a moment, I would like to point out that I am a big detractor of much of the so-called "educational" games that have invaded the market. As I pointed out last time, much of the software I've seen, are merely games thinly disguised as learning tools.

Traditionally, educators can't program, and programmers can't educate. Not so with Fisher-Price, as these two games are wonderful. Both of them are simply, but fully illustrated — similar in style to a child's storybook — and involving enough to hold the interest of kids.

Beginning with **Peter Rabbit**; the game offers users a demo mode, and four skill levels. While adult supervision is probably a good idea (anyone who lets their 3–6 year old loose on their home computer deserves what's left when their offspring is done with it), it is not really necessary during the operation of the game itself. The instructions are simple enough for a child to understand and perform unaided. Played with either a joystick or directly on the keyboard, the child moves Peter through a meadow and over several bridges to arrive home.

At the onset, Peter is alone in a field. By pressing the space bar or moving the joystick, Peter will hop along his way. As he approaches the edge of the screen, Peter will exit from view, and the drive will access the floppy to retrive the next screen (resulting in a slightly annoying 10–12 second delay in the game).

As Peter moves throughout the field, he will pass other inhabitants (a couple of mice, a dog and others). Each of these animals will be standing next to a letter of the alphabet (the mice near an "M" the dog by a "D", etc.) Also in the field are five "special" friends of Peter.

Each of the "special" friends stand on the other side of a barrier (a chasm, field of corn, river). The child (in the guise of our fuzzy-eared hero) is then required to play a game in order to pass. When a correct answer is given, Peter's friend nods his head "yes", a note is sounded, and an object is removed, or a section of a bridge is built. In the event of a wrong answer, the friend shakes his head "no" and a deeper tone is sounded. The child has as much time as he or she needs to answer each question, and is is not penalized for wrong answers.

These games consist of (depending on the difficulty level) sound to symbol matching, upper/lower case letter matching, consonant matching with and without voice, and consonant blends in short and long vowel words.

All of the games/lessons are played with the child given four choices from which to choose an answer. Answers are, again, chosen with either the joystick or the keyboard. After the bridge is built or the objects removed, Peter can pass on his way. At a couple of points in his travel, Peter is

given an option in paths. Should he choose the wrong path, a stone wall will bar his way, requiring him to turn around and retrace his steps. Once Peter has finally reached his home, his mom will come outside and give Peter a present.

First Men is similar in structure to **Peter Rabbit**, in that the child can run the program almost without assistance. In this game, the user is represented by a spacesuited man (the "Professor") in a flying saucer. The ship is "fueled" by solving math problems, and navigated via the joystick.

Your astronaut leaves earth and maneuvers through an asteroid field (solving numeric problems along the way to remain in flight). When fuel runs low, the child must land on an asteroid or he or she will drop (albeit slowly) to the ground.

Upon reaching the moon the Professor must land, exit the ship, and proceed to the underground caverns of the Selenites. This underground cavern consists of a series of caves connected by descending ladders, and populated by odd looking creatures called Selenites. Upon approaching a Selenite, the Professor will be stopped and asked to solve a word problem ("If I have 100 sculptured rocks in my possession, and the person with the most has 350, how many do I need to have to have the more than him?").

A correct answer will banish the Selenite and net the Professor a lump of gold. Unlike **Peter Rabbit**, **First Men** keeps score, with 200 points acquired each time the Professor picks up a piece of gold. The child wins the game when he or she finds the Grand Lunar and then escapes back to the ship.

As stated, both of these educational products are from Fisher-Price Learning Software, P.O. Box 1327, Cambridge, MA 02238 (phone, 617-494-1222). As an added plus, Fisher-Price is currently offering a "Buy three, get one free" offer. Details are included in specially marked packages, and applies to all Fisher-Price, and sister company's software (Spinnaker, Windham Classics, Telarium, Better Working). The offer runs through to January 31, 1987.

Next up is a personal favorite of mine, **Superman, the Game** from First Star Software. This is not only Superman's second outing in the electronic games market (the first being the delightful Atari 2600 game) it is the First Star's initial offering in their **Super Powers** series, which is licensing DC Comic's superheroes (the next should be Wonder Woman).

Like its predecessor, this incarnation of Superman is a fast-paced, all-action, thrill-a-minute game. The premise of the game is simple, Krypton's native son must protect several innocent citizens from the evil and villanous Darkseid (pronounced "Darkside"). This joystick-powered game consists of 11 different screens (six map screens, and five battle screens).

First, the map screens: Three of them are an overhead view of the streets of Metropolis, and three are deep within Darkseid's lair. In these screens, Superman and Darkseid race around attempting to herd several bystanders either to safty or Darkseid's prison, from which there is no escape. Placed at various junctions throughout the streets of the city (and the by-ways of the netherealm) are deflectors which can be used by the two combatants to angle the blast of their heat vision (Superman) and Omega Beam (Darkseid) to strike at their enemy.

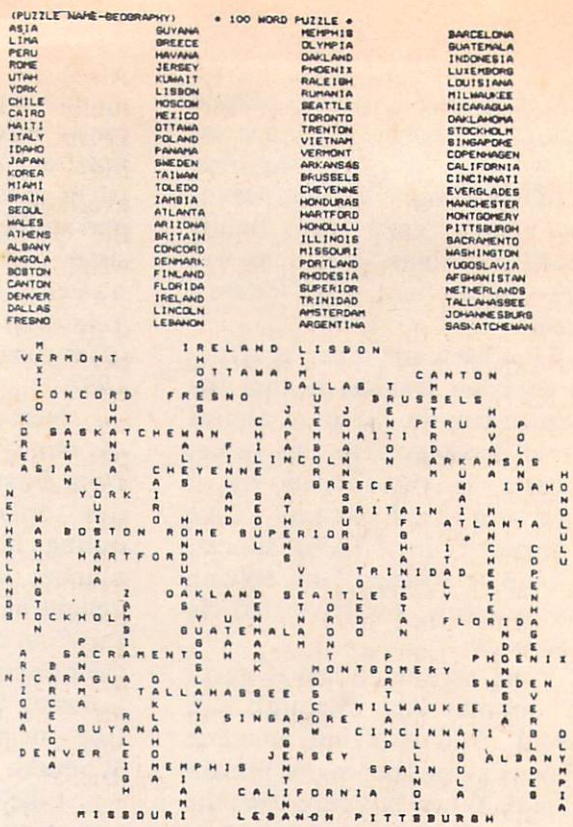
When struck by the other's power blast, each will lose points and power. Beams will not injure citizens, but will push them in the direction of the beam (Darkseid's beam will transport them to his relm). By utilizing the deflectors both hero and villain can "bend" their beams around corners, *and* dodge oncoming blasts.

Connecting the various map screens are the battle screens where Supes and Darkseid go head-to-head in their eternal struggle. These battlegrounds include an extended corridor through which Superman must fly, attempting to evade missles, and capture his foe, a second corridore where the situation is reversed, and Darkseid is chasing Superman, one where The Man of Steel must use his super breath to blow fireballs back into a circle of cannons, another where he must stop a barrage of bombs from being dropped on his adopted city. The fifth screen is a descending well where Superman must dodge bombs and destroy Darkseid's mobile launcher. This well connects the fair city of Metropolis with the netherworld of Darkseid's dominion.

Here is a game that will thrill for hours gamers young and old. A warning here, it is not easy to master this game, as the action runs along at a fever pitch (there are skill levels, and an option on whether or not you want the battle screens), and the action is non-stop.

Remember, these are the folks who gave us two wonderful **Spy Vs Spy** games, and Superman is, as one comic book professional recently told me, "The character that created, and defined the superhero genre." Considering all this, **Superman the Game**, is a sure-fire winner. First Star Software, 18 East 41st Street, New York, NY 10017. Phone: (212) 532-4666.

Our fourth product is (surprise Randy) the **Stack Light Rifle** from Romaro Enterprises International, Limited, a Canadian firm. This device works much akin to a



THE PUZZLE GENERATOR

THE PUZZLE GENERATOR is a complete Criss-Cross and Word-Search Puzzle development system for your Commodore 64 and 128 (in 64 mode) computers. It utilizes more than 15 built-in word categories to give it the capability to generate BILLIONS of puzzles, all automatically.

This powerful program diskette contains many features: Criss-Cross puzzles can be printed with or without a starter word; By varying the grid, puzzle size can range anywhere from 2 words to 100 words; Built-in word categories include Railtalk, Games, Boys and Girls Names, Fun Things, Computers, Adventure, Chess, Football, Baseball, Geography, Good Book, General Interest and more; Word editor enables users to create special interest puzzles from any list of words; including most foreign languages; Works with any printer (required); Puzzles, Answers, and Word Lists that you create can be saved on diskette; Menu driven for easy operation, and much more. Armed with these features **THE PUZZLE GENERATOR** transforms the worlds number one computer into the NEW King of Puzzles!

THE PUZZLE GENERATOR is a program that will enhance anyones education and is now being used in many schools throughout the USA. **PRICE $34.95**

Data Disk for above with over 100 additional word categories (not required). **PRICE $10.00**

30 DAY MONEY BACK GUARANTEE

Place your order today to receive **THE PUZZLE GENERATOR.** To place your order, please fill out the following information:

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____/_____

PAYMENT *(CIRCLE ONE)*: VISA MASTERCARD CHECK MONEY ORDER C.O.D.

CARD NO# _____ EXP. DATE _____

Includes shipping and handling. PA residents add 6% sales tax.

ALSOFT • 305 LARGE AVENUE • CLAIRTON, PENNSYLVANIA 15025 • **(412) 233-4659**

light pen, with the rifle activating the game. The rifle works with both the 64/128 and the now defunct VIC-20 (as if anybody cares). With it the user can stand on the other side of the room (some 10–12 feet away) and shoot directly at the screen at the attackers. While this sounds real nifty (I know it did to me at first) the game is not without its problems.

The first of which is that the provided software is super-dull. There's no flash to the eight games on the disk, you simply stand on the other side of the room and shoot at the screen (I've played 2600 games that were more exciting than some of these). The gun's sight has no cross-hairs to line up on the target, and the gun itself tends to fire high and to the left — at least mine did.

Finally, to move from one game to another, you must power down and re-boot the disk. Not so easy from 12 feet away. Why a method wasn't provided to cross-over between games is beyond me. Also, a couple of the games actually require the user to access the keyboard to leave high score initials. Two additional games are available, both being better than the provided ones, but not by a whole lot.

Still, I don't want to be too down on the product, as the experience wasn't entirely without enjoyment. After all, there is a certain charm to playing a computer game from across the room. I forgive the manufacturer for the lack in the programming, as it is a first effort, and breakthrough technology doesn't have to be breathtaking. With any luck, and a serious programming effort, the next generation of games should serve up the type of gaming for which folks like us are looking. Romaro Enterprises International, Limited, P.O. Box 227, Streetsville Postal Stn., Mississauga, Ontario, Canada L5M 2B8.

One last item (also by way of suprise) I've been using a new GE monitor the past few months, (especially while reviewing these games) and it's wonderful! This sucker is as much better than a 1702 than a 1702 is over a regular TV set. And the best part is that the GE monitor *is* a TV set. This 13″ color (or 12″ B&W) monitor is a super TV set as well being one super monitor. A sharper on-screen image you couldn't get. I highly recommend, if you're in the market to replace (or acquire) a monitor, that you shine on the 1702 and pick up this GE unit. (As an aside, GE is also manufacturing a modem, printer, and data cassette, that also exude hi-tech GE care and production, I'd suggest checking out the entire line.) GE, Electronics Park, Building 5, Syracuse, NY 13221. Phone 1-800-626-2000.

Well, I guess that's about it for this time around, hope it's been as much fun for you as it's been for me. Catch you next time around, eh?

■

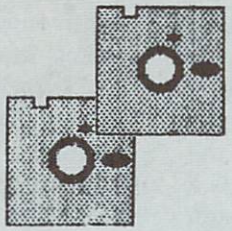
PREMIUM BULK DISKETTES

COMPUTER MA

2700 N.E. Andresen Rd.
Vancouver, WA 98661
(206) 695–1005

Colored — 79¢ each

- Red
- Green
- Gray
- Light Blue
- Dark Blue
- Lavender
- Burgundy
- Brown
- Yellow
- Orange



Black — 59¢ each

- Lifetime Warranty
- Reinforced Hubs
- Tyvec Sleeves
- Over 300,000 Sold!

ACCESSORIES

Disk Notcher	$ 5.99
Storage Box 100 Count	$15.99
Amaray Disk Bank (5) Holds 50	$10.99
C-64 Power Supply Repairable	$39.95
4-Slot Expansion Board	$29.95
Mach 128	$37.99
Mach 5	$26.99
Epyx Fast Load	$29.95

Pico Command Center —

- Tilt Monitor Base
- Surge Protected
- 5 Lighted Switches
- #1 Best Seller
- UL Approved
- EMI Filtering

$59.99

SAVE $20.00

RIBBONS

Star SG-10	$ 1.99
CBM 1525	$ 7.99
CBM 1526	$ 7.99
CBM 801	$ 7.99
Panasonic 1091	$ 9.99
Star SG-10 Colored	$ 3.29
Epson MX or RX 80	$ 6.99
Commrex 220	$ 7.99

Choose from HUNDREDS of products for the

Commodore 64 & 128

LOW DISCOUNT PRICES

MW-350 INTERFACE

$74.99

★ With 10K Buffer

★ #1 Printer Interface On The Market!

Mail Order Available

VISA COD MasterCard

Same Day Service

Open 7 Days a Week:

Mon–Sat.....9–8
Sunday.....12–5

"MD + 64" $9.99

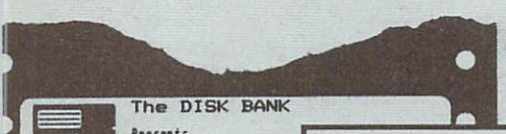
The PERFECT Utility Program for hackers only.

- **View any track or sector through 35**
- **Prints out sectors**
- **Change or copy sectors**
- **Decode the tokens**
- **Decode the BAM**
- **Not Copy Protected**

Pepper's GENEALOGY — OR — DOG HISTORY Program

Family tree up to 5+ generations
Prints out
Unlimited number of files
Saves to disk
C-128 (80-column) version OR C-64 format available
Not Copy Protected

YOUR CHOICE $29.95



NEW!

ANNOUNCING...

OUR FIRST OFFERING.....

The GRAPHIC LABELER
$17.95
Using Printshop Graphics create your own mailing labels, disk labels, etc.

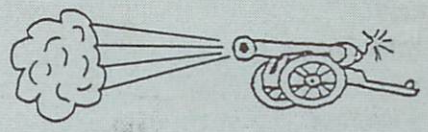
FEATURES INCLUDE:
* Up to 6 lines of text.
* 10 Different Fonts.
* Use of Mixed Fonts.
* Prints Multiple Labels.
* Uses Standard Labels.
* Works with most Printers – Star, Citizen, Panasonic, Epson, etc.

COMMODORE MPS-801, 1526 VERSION LATER.

ACTUAL SAMPLE

KRACKER JAX

PROTECTION BUSTERS



Blast Disk Protection

The Next Generation of Copy Utilities Is Here Today!!

*** * * KRACKER JAX * * ***

A REVOLUTION IN ARCHIVAL SOFTWARE !

This exciting new software begins where all other copy utilities leave off — Kracker Jax not only backs up your expensive programs, it also completely strips all protection in the process. You end up with a copy that is TOTALLY BROKEN!

And, believe it or not, that's only the beginning. Check out some of these exclusive features:

- **Kracker Jax is the ONLY program of its kind!**
- **Kracker Jax requires NO special knowledge to operate!**
- **Kracker Jax will back up OVER 80 separate titles!**
- **Kracker Jax strips protection in a matter of SECONDS!**
- **Kracker Jax is a GREAT BARGAIN — just $19.95, complete!**
- **Kracker Jax is UNPROTECTED — easy to back up!**
- **NEW Kracker Jax disks are available regularly!**

Kracker Jax is a parameter copy system. Each Kracker Jax disk contains over 80 parameters — and each parameter backs up a specific title. What's a parameter? Just a set of instructions that allows your 1541 or 1571 disk drive to strip all, and we mean ALL, copy protection from your expensive software, leaving you with UNPROTECTED back-ups that can be copied with anything!

This system has many advantages over older "nibble" type utilities. For one thing, you don't have to experiment. Each parameter WILL backup the title it was created for. Period. For another, a back up created with Kracker Jax will NEVER rattle your disk drive head.

Kracker Jax is one of the hottest new products of 1986! If you want to protect your investment in your expensive software, there is simply NO BETTER utility on the market!

Only

*** * * $19.95 * * ***

Complete !

More Computer Magic
Mystic Calendar

by John Olsen

Much has been made of time travel in movies and literature. Time seems to possess a magical quality. Some of the mysteries of time are examined in this magic trick for your Commodore 64, called the Mystic Calendar.

Imagine that you were permitted to travel forward or backward in time to any month of your choice. Further imagine that you were allowed to visit one day out of each week during that month. Which days would you choose? And what month during what year? Would you choose December 7, 1941 (the date that will live in infamy)? Perhaps attend the wedding of the century? Or travel to one of your future birthdays? You are free to pick any year and month in this computerized magic trick.

You can choose to travel to the past, present, or future by giving the computer any year and month. The computer will present you with a calandar for that month, and ask you to pick one day out of each week. And it will ask you only how many Sundays you picked, how many Mondays, Tuesdays, etc. You never give it a clue as to exactly which dates you chose. And to make things more difficult, some months have four weeks, some have five, and some even overlap into six weeks. You even had a free choice of years and months. Yet, without hesitation, the computer will tell you the sum of your dates!

Let's review that, to impress upon you just how impossible this really is. You had chosen a year and month. The computer showed you that month's calendar; no fooling around here, the calendars are 100% historically accurate. You picked one day out of each week; any day you preferred. You could have picked all Mondays, some duplicates, or all different days of the week. There were no restrictions here. The computer asked you how many Sundays you had chosen. Then how many Mondays. Then Tuesdays, Wednesdays, and so on through the end of the week. That is the *only* information you gave the computer. You gave no indication whether the Sundays were near the beginning or end of the month; likewise with the other days.

Yet with only this seemingly irrelevant information, the computer could divine the sum of the dates you chose. If you try it again with a different calendar or even the same one, the results will always be different. But yet each time the computer can unfailingly determine the sum of the dates you picked. A most astonishing effect!

To try this trick, you must type in the program listed here. Be sure to type it in exactly as shown, or it may not work properly. Be especially careful with the keyboard graphics characters found in lines 250-280, line 330, and line 350. They draw the calendar and need to be entered carefully.

A description of the workings of this trick and some programming techniques will follow. It is not necessary to understand these to type in, run, and enjoy this trick. You can stop reading now and begin entering the program. But, if you want to know how the trick works, and how the programming assists the trick, the rest of this article will most likely make it a little clearer.

In order for this trick to work, the magician needs to see the calendar that was chosen. Your computer doesn't have eyes, so you can't choose the calendar hanging on your wall. If you did, the computer wouldn't be able to see it and would not be able to do the trick. So the computer produces the calendar for you, right on your TV screen. Although it appears that this is done for *your* convenience, so that you can choose any date in the past or future, the real reason is so that the computer can see the calendar.

In simple terms, the computer adds the dates of all the Wednesdays as it creates the calendar. Then when it asks for the number of Sundays, Mondays, etc., it adds or subtracts the number of days between that day and Wednesday. Specifically, the computer will add all the Wednesday dates together. If you tell the computer you picked one Sunday, it will subtract three from the total, because Sunday is three days

before Wednesday. If you tell the computer you picked three Mondays, it will subtract two, three times. This is because Monday is two days before Wednesday. It subtracts one for each Tuesday, and, naturally, zero for each Wednesday. The computer adds one for each Thursday, two for each Friday, and three for each Saturday, because they are one, two, and three days after Wednesday.

So, the secret is really as simple as adding the Wednesday dates together, and then adding or subtracting each time a day of the week was chosen. But as you can see, the magician needs to be able to see the Wednesday dates. The computerized magician accomplishes this by creating the calendar for you and adding up the Wednesdays as it draws them on the calendar. Then it adds or subtracts as it asks how many Sundays, Mondays, etc.

Programming this trick into a Commodore 64 is broken down into two main parts. The first is to accurately create a calendar for any date. The second is to ask the seven questions and figure out the correct sum.

Creating a calendar seems difficult at first, but, after a little thought, becomes simpler. You start by finding out what day the first of each month in 1900 was. January first was on a Monday, February first was on a Thursday, etc. Each year, this shifts up a day. So in 1901, January first was on a Tuesday, February first was on a Friday, etc. Knowing this makes things a lot easier. Just find how many years have passed since 1900 and add on that many days. And don't forget the extra days for the leap years. Doing this will give you a simple way to find the first day of the month.

Once you have the first day of the month, it is easy to program the computer to print out the dates, stopping at the end of the month. This information is stored in the data statements found in lines 180-200. The computer reads the data statements for the name of the month, the first day of that month, and the total number of days in the month.

Using the first day of the month (from the year 1900) the computer determines the first day for the month and year you picked. This is done in lines 210-230. It prints the title in line 240, and prints out the calendar itself in lines 250-350. Notice that while printing out the calendar, it keeps a running total of the Wednesday dates in line 320. This total is saved in the variable **Q**.

The calendar is printed in black onto a black screen, so that it can't be seen as it is being created (a task that takes about 2 seconds). Then the screen colors change to white and the calendar appears instantly! Likewise, instructions were printed in black in between the previous lines, and they, too, now appear instantly.

After the person who is watching the trick (the trickee?) has picked the dates, he presses a key and the calendar scrolls up to the top of the screen. Because each calendar can be anywhere from four to six weeks long, the screen can't scroll a set amount. Line 370 checks to see when the calendar has reached the top of the screen and then stops scrolling.

At this point the seven questions are asked. This is done inside a loop, where the same question is asked each time, with only the name of the day changing. The names of the days are held in the **D$** array, and are printed out in the question shown in line 440. Line 450 adds or subtracts the correct amount from the Wednesday total held in the variable **Q**.

The last thing to do is to announce the total, after an appropriate build-up. Line 480 makes

the screen colors change rapidly, as if the computer were thinking. Then the sum of the dates is revealed in lines 490 and 500, and the trick is completed!

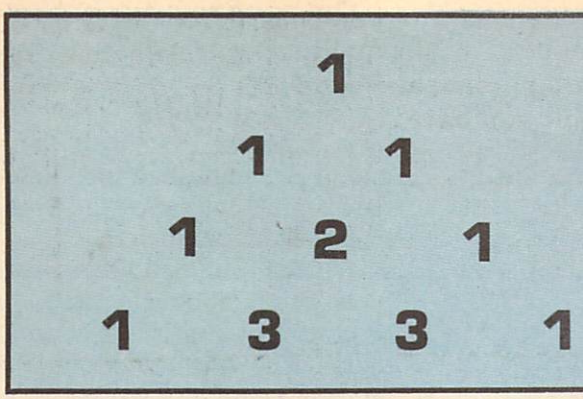
As you can see, converting a magic trick from one that a person can do to one that a computer can do, requires careful planning. A computer has no eyes to see a calendar held by a spectator, so we must compensate for this fact. But a computer has the ability to calculate much faster than a human, and in this lies one of its strengths. I hope you have enjoyed this trick and can enjoy watching it puzzle your friends. Until next month, *happy conjuring!*

■

Mystic Calendar Listing

by John Olsen

```
10 poke53280,0:poke53281,0:printchr$(142
)chr$(8)"[clr]";
20 print"[blk]  below is the calender fo
r the month."
30 print"[wht]mystic calender          by
 john olsen"
40 print"[blk]   pick one day out of eac
h week, and"
50 print"  write the dates on a piece of
 paper."
60 print"[wht] you can choose any month
during the"
70 print" twentieth century (past or fut
ure)."
80 gosub510
90 input"[up][wht] pick a year (1900-199
9)";y$:y=val(y$)
100 ify<1900ory>1999then90
110 y=y-1900:print
120 input"[up] now pick a month (1-12)";
m$:m=val(m$)
130 ifm<1orm>12then120
140 forx=0to6:readd$(x):next
150 datasundays,mondays,tuesdays,wednesd
ays,thursdays,fridays,saturdays
160 forx=1tom:readm$,d,t:next
170 datajanuary,1,31,february,4,28,march
,4,31,april,0,30,may,2,31,june,5,30
180 datajuly,0,31,august,3,31,september,
6,30,october,1,31,november,4,30
190 datadecember,6,31
200 s=int(y/4)+y+d
210 ify<>0andy/4=int(y/4)andm<3thens=s-1
:ift=28thent=29
220 s=1-s+int(s/7)*7
230 m$=m$+str$(1900+y):printspc20-len(m$
)/2)"[blk]"m$
240 printtab(9)"[cmdr-a][shifted *][shif
ted *][cmdr-r][shifted *][shifted *][cmd
r-r][shifted *][shifted *][cmdr-r][shift
ed *][shifted *][cmdr-r][shifted *][shif
ted *][cmdr-r][shifted *][shifted *][cmd
r-r][shifted *][shifted *][cmdr-s]"
250 printtab(9)"BsuBmoBtuBweBthBfrBsaB"
260 printtab(9)"[cmdr-q][shifted *][shif
ted *][shifted *][shifted *][shifted *][
shifted *][shifted *][shifted *][shifted
 *][shifted *][shifted *][shifted *][shi
fted *][shifted *][cmdr-w]"
270 printtab(9)"B";
280 forx=0to6
290 ifs<1ors>tthenprint"  ";:goto310
300 printright$(str$(s),2);
310 ifx=3thenq=q+s
320 s=s+1:print"B";:nextx:print
330 ifs<=tthen260
340 printtab(9)"[cmdr-z][shifted *][shif
ted *][cmdr-e][shifted *][shifted *][cmd
r-e][shifted *][shifted *][cmdr-e][shift
ed *][shifted *][cmdr-e][shifted *][shif
ted *][cmdr-e][shifted *][shifted *][cmd
r-e][shifted *][shifted *][cmdr-x]";
350 poke53280,1:poke53281,1:gosub520
360 ifpeek(1073)<>112thenprint:goto360
370 print"[up][up][up][up][up][up][up][u
p]"
380 print"[pur]add together the dates yo
u wrote down on";
390 print"the paper.  i'll wait...[down]
[down]"
400 gosub510:gosub520:print"[up][up][up]
[up][up][up]";
410 forx=1to5:print"
                    ";:nextx
420 poke198,0:forx=0to6:poke53280,x+2:po
ke53281,x+2
430 print"[home][down][down][down][down]
[down][down][down][down][down][down][dow
n][down][down][down][down][down][down][d
own][down][blk]how many "d$(x);
440 input" did you pick";z$:z=val(z$):q=
q+z*(x-3)
450 print"[home][down][down][down][down]
[down][down][down][down][down][down][dow
n][down][down][down][down][down][down][d
own][down]
          "
460 fory=1to200:nexty,x
470 forx=0to99:poke53280,xand15:poke5328
1,xand15:nextx
480 printtab(10)"[up][up][up][blu]the da
tes you picked"
490 printtab(13)"[cmdr-2]added up to"q
500 goto500
510 print"[blk]press a key when you're r
eady to go on.[down]":return
520 wait197,64:wait197,64,255:return
```



PASCAL'S TRIANGLE

by Carmen Artino

Public Key Cryptography

Consider the following "secret" message:

VHQG KHOS

The message is secret because we can't read it; it has been subjected to some coding procedure that renders it unreadable to all but it's intended receiver. If we knew the coding scheme, we could reverse it to render the message readable. Now, this particular message was encoded using a substitution cipher. A substitution cipher is nothing more than a shifted alphabet; the above message was encoded using the following key, it's called the Caesar Cipher:

```
A B C D E F G H I J K L M N O P Q R
D E F G H I J K L M N O P Q R S T U

S T U V W X Y Z
V W X Y Z A B C
```

To encode or encrypt a message, find a letter in your message in the 1st alphabet. It's coded form will be the letter beneath it in the 2nd alphabet. Thus, an "A" in the message to be encoded will become a "D" in the coded message. To decode (decrypt) the message, simply reverse the procedure. Now we see how easy it is to decode the above message; it says, "SEND HELP". Of course, if the key was unknown to us, the above message would be only slightly more difficult to decode; substitution ciphers are relatively easy to break!

We have used this rather simple example to demonstrate one of the main problems in transmitting messages whose content is to be known only to the intended receiver; namely, both the sender and the receiver had to know the key. The key was to be used to encode and to decode the information.

There have been some very clever and ingenious schemes for coding/decoding keys developed throughout history and many have remained secure (unbroken) for a long time. Others have been broken rather quickly; for example, the Japanese code during World War II was broken soon after U.S. entered the war. But all of these codes still shared the problem just mentioned. The recipient had to have knowledge of the key.

Is there an "unbreakable" method for encoding and decoding? Perhaps not, but I would like to present a method that comes close; better yet, it does not suffer from the problem we just discussed. The recipient does *not* have to know how the message was encoded to render it readable; in fact, the recipient need not know the sender personally.

The method was developed fairly recently by three mathematicians, Rivest, Shamir, and Adleman, and has become known as the RSA method. The theory on which the coding/decoding scheme is based, however, goes back 300 years and is based on a simple result in a branch of mathematics called Number Theory. The result was discovered by Pascal's friend and colleague, Pierre de Fermat.

We shall describe a greatly simplified version of the RSA method which contains all the flavor of the full version but is much easier to implement. The full version would require a very fast and powerful computer whereas the version we shall describe can be implemented using a Commodore 64 or even a pocket calculator.

The method has also become known as public-key cryptography because each person who is allow-

ed to participate in the message game has an encoding key which is placed in a public directory of such keys something like a telephone book. If Tom wants to send Dick a message without Harry or anyone else knowing what it says, then all Tom has to do is to look up Dick's encoding key in the public directory and encode the message he wished to send to Dick.

The knowledge of how the message was encoded (which is, of course, publicly available via the directory) is of little help to Harry if he trys to decode it. Only Dick will know how to decode the message he receives!

Let's describe the method. First, we agree on some scheme to assign numbers to the letters of the alphabet; for example:

A=02, B=03, C=04, . . . , Z=27

We shall see in a moment why it is not desirable to have A=01. Now, we pick a prime number larger than 27 — say 29.

Next, the public encoding key will be some number between 1 and 28 (29 - 1), call it **e**, having the property that it and 28 have no common factors. Let's choose **e = 5** for our example; this is the key that would be published in the directory. If Tom now wished to send Dick a message, say, "ACT NOW", then Tom would proceed as follows:

First, he would change each letter in the message to its corresponding numerical equivalent. Thus the word "ACT" would be 02 04 21. Now Tom looks up Dick's encoding key in the directory and finds 5. Next, Tom computes the 5th power of each letter and reduces it modulo 29.

This means that after the 5th power of the letter is found, it is divided by 29 and the remainder is kept. This is precisely the MOD function in Pascal: **m MOD n** returns the remainder when **m** is divided by **n**. In our example, we convert the "A" in "ACT" as follows:

$(02)^5$ MOD 29 = 32 MOD 29 = 03

(29 divides into 32 1 time with remainder 3; it is the 3 we are interested in.) Thus the "A" encodes into "B". When we continue this process, we find

$(04)^5$ MOD 29 = 09
$(21)^5$ MOD 29 = 02
$(15)^5$ MOD 29 = 10
$(16)^5$ MOD 29 = 23
$(24)^5$ MOD 29 = 07

Thus, the encoded message becomes BHA IVF. (If A = 01, then any power of A would still be 1, and 1 MOD 29 is again 1. Thus, "A" would encode into itself if we assigned 1 to A).

Tom, of course, is quite pleased. But how does Dick decode this message? Dick has his own decoding key which, of course, he *does not* publish. This key is determined by finding the number **d** with the property that **e*d MOD 28 = 1**. That such a unique number always exists is a consequence of Fermat's result mentioned above.

When **e = 5**, we find that **d = 17** because **5*17 = 85**. When we divide 85 by 28, we get 3 with remainder 1. Therefore, 17 would be Dick's decoding key and he would follow the same procedure that Tom did to encode the message. Thus, to convert the **B** back to its original form, Dick would compute:

$(03)^{17}$ MOD 29 = 02

so that the **B** converts back to **A**. With a few more computations, Dick would then be able to read Tom's message.

The privacy of a message in this system rests upon preventing an unintended recipient from learning the value of **Modulus**; in our case, 29. If this value is known, then, since **e** is already known, a fairly simple calculation will produce the value for **d**.

In the full version of the RSA scheme, the modulus is replaced by a number which is *not* a prime but is the product of two "very large" primes, let's call them **p** and **q**. The published encoding key would then consist of the **product p*q** and the number **e**.

If the prime numbers **p** and **q** have about 50 digits each, their product will consist of about 100 digits. Knowing the product **p*q** and the value of **e** in this version would be of no help in finding **d**, the decrypting key. In order to determine **d** from the known values, *we have to know the values of p and q!*

Recall from last month's article that the process of finding the factors of a number is called *factoring*. With the product of **p** and **q** being a number of about 100 digits, the factoring process could take a very, very long time even using the fastest algorithms on the fastest computers. (The procedure we presented last month does *not* fall into the category of fast algorithms.)

However, some recent developments have lead reseachers in this field to reconsider the RSA public-key method. In fact, in 1984 a 76 digit number was factored on a Cray-1, a very fast computer using a new algorithm, in 32 hours, 12 minutes! This is a remarkably short time; before that, it was estimated that a 100 digit number would require about 75 years to factor! It is the nature of mathematics that what is unattainable now becomes attainable "shortly". This has been true throughout its history. It is thus a safe prediction to say that the RSA scheme will shortly be a method of the past.

The author welcomes comments and suggestions concerning this column. The interested reader may write to the author at P.O. Box 43, Guilderland, NY 12084

Potpourri

Quickies, Short Takes & Nutshell Reviews

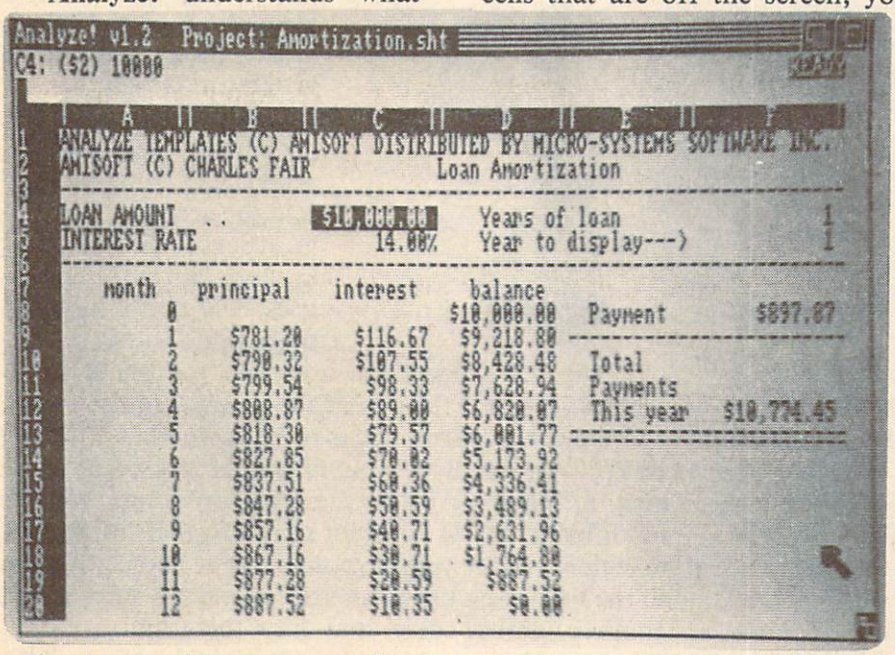
Analyze! is the first spreadsheet program for the Amiga. I was wary as I pushed in the disk; the first group of software for a new machine is usually as disappointing as the machine is exciting. What a surprise. Let me offer my praise right off — there is nearly nothing wrong with this product. It claims to be a spreadsheet, and it is just that.

Analyze! breaks no new ground as a spreadsheet. It simply works well with no fuss or muss. The command structure is all but generic, so, if you have worked with spreadsheets before, you will find **Analyze!** a comfortable place to work right from the start. Best of all, the program itself seems to have found a comfortable place for itself within the Amiga. It can be run from the Workbench, and is willing to work within the restraints of window placement and size.

Analyze! understands what the creators of the Amiga had in mind; it knows how to co-exist in a multi-tasking world. (It does need a little help from you, though. The program is capable of handling a 8192 row by 256 column sheet of cells, provided you're not going to have other needs for the memory. On start up, it stops to ask you how much memory you're willing to let it have.)

Natural and convenient extensions follow from this combination of application and machine. For example, in addition to the usual typed (or cursor direction key) indication of cells, you can move about with the mouse. Somebody sensible designed this thing. Each control element is used where it applies best and no further.

The mouse can point to cells or even ranges of cells, but only the ones that are visible on the screen — fast and easy. But, for cells that are off the screen, you will have to move back to the keyboard.



Analyze! spreadsheet screen showing one of the templates (usable examples) found on the program disk.

Analyze! by Micro-System Software, Inc. is a comptetent and fairly-priced ($99.95 list) piece of work.

Grant Johnson

Heart Of Africa

— Electronic Arts

Sequels seldom live up to the standards of the original, but in **Heart of Africa** the Bunten brothers have taken the original concept and design of **Seven Cities of Gold** a full evolutionary step farther. Based upon the same principals of mapping and exploration of a piece of geographical history, you can now journey into the darkest reaches of the unknown African frontier.

Adding more features and levels of complexity than can be adequately addressed in a capusule review, this package offers enough new options and details to make it far more than just another **Seven Cities of Gold**. The concept is familiar enough that you will start out feeling comfortable in the playing environment, but there are enough new twists and strategies to master that you won't take anything for granted.

To truly appreciate this one, curl up in front of your monitor and let the creators of **M.U.L.E.** take you into another world. You'll be surprised as you notice that the morning sun is crawling over the horizon, and you still have most of the unexplored continent awaiting you. Fortunately, you can save the game and continue later.

Randy Chase

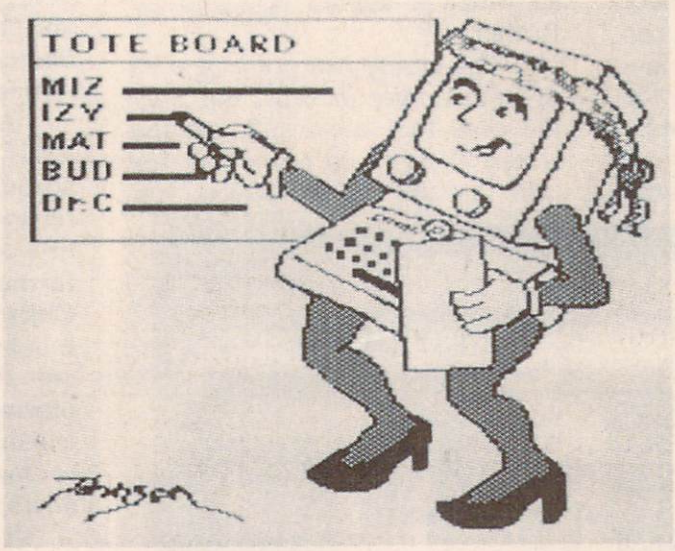
The Bud Izzit Art School

Let's Get Behind Bars

by Eddie Johnson

Editor's Note: After hearing about Senator Jake Garn's trip on the Space Shuttle, Bud has dropped his plans to run for the House of Representatives, and is busy working on his Senatorial campaign (he is still determined to be the first artist in space). Since he is currently in New York having his image worked on by Madison Avenue media specialists, we have asked staffer Eddie Johnson to fill in for him again.

If you have been following this column for the past couple of months, you may have spotted some screen formatting tricks that we have been using to imitate the PRINT locating type commands that you encounter in some dialects of non-Commodore BASIC. This month we're going to explore these techniques a little further, and look at ways to translate programs from one computer to another. Also, in our "Art Gallery" this issue, we have a few little circle programs using our circle formula from "Izzit #1" (in the October, 1985 issue of *The Guide*, pages 50 and 51).



The cursor positioning commands `PRINT AT X,Y` (or, `PRINT@X; LOCATE X,Y`, etc.) and `HTAB(X): VTAB(Y)` that you see in some versions of BASIC allow you to choose in which row (X) and column (Y) you want to PRINT your next character. These handy little tools are missing from PET BASIC, but we Commodore users have the unusual ability to write our cursor controls directly into a character string.

If you look at the first two programs, "FOR MAT" and "64MAT," you will see in line 100 how we build a "down" cursor string and a "right" cursor string which are then used in line 140 to position our LINE NUMBER PRINT statement.

Because the VIC and the 64 have different screen formats (the VIC has 22 columns and 23 rows, while the 64 has 40 columns and 25 rows), the formulae in lines 140, etc., are also different. To get some practice in translating between different machines, study both programs and see if you can figure out why the VIC 20 program will work in the C-64, while the C-64 program will *not* work inthe VIC. When I compared them myself, I suddenly realized that I could combine some of the ideas from both programs and make a bar graph program!

Have you been catching flack from your spouse or your parents (or your children) for "wasting time" doing "frivolous" graphics programs? Well, here's one that will make them think you're actually doing something serious, for a change. (Just don't let them watch too closely!)

After you RUN the programs, you will be asked for an input and told the accepted range of numbers. Experiment around with different entries, and don't forget to press [RETURN].

The VIC 20 version asks you for the INCREMENT, which is the size of the "steps" in the graph, and the C-64 version wants to know the MAXIMUM INPUT, or the largest number that you want to be able to enter. Again, the two programs can be modified to fit the different screen formats of the two machines. After you have made your choice, we flip to the bar graph page, where you make a letter entry, another number entry, and draw the graph.

In addition to a "down" cursor string and a "left" cursor string, we also have an "eraser" string

and a "bar" string. Line 110 is a very busy line! The "eraser" string permits you to change your mind and alter the entries as many times as you wish.

You will notice that the graph is not terribly accurate (*i.e.*, 95 is the same length as 90, etc.) but how much sophistication do you expect from a six-line program? If you wanted to get creative and use the keyboard graphics characters that make different width vertical lines, I'll bet you could divide the last space in the bar to get more sensitivity in the graph. In fact, let's use that as a beginning suggestion for . . .

Challenge #5

Try to improve upon or modify the bar graph. How about using POKEs instead of PRINTs? Can you make the bars vertical rather than horizontal? Do you know the POKEs and PEEKs for the cursor location? Try using different colors or different symbols. Be sure to send us your best efforts. But, *be careful!* Instead of being accused of "frivolity", now you're going to get yelled at for hanging out in the bars!

■

"FOR MAT" Listing

by Eddie Johnson

```
1 rem **** program 1 ****
2 rem ***** for mat *****
3 rem ***** vic  20 *****
100 d$="[home]":fors=1to22:d$=d$+"[down]
":r$=r$+"[right]":next:r$=r$+"line numbe
r"
110 print"[clr]";:forl=1to22:printl:next
120 print"[home][rvs on]"right$(r$,15)"
   [left][left][left][left]";:inputn
130 ifn<2orn>22then120
140 printleft$(d$,n)right$(r$,13+int(n/1
0)+1)n;:goto120
```


"64MAT" Listing

by Eddie Johnson

```
1 rem **** program 2 ****
2 rem ****** 64mat  *****
3 rem ****** c-64  ******
100 d$="[home]":fors=1to25:d$=d$+"[down]
":r$=r$+"[right]":next:r$=r$+"line numbe
r"
110 print"[clr]";:forl=1to24:printl:next
120 print"[home][rvs on]"right$(r$,22)"
   [left][left][left][left]";:inputn
130 ifn<2orn>24then120
140 printleft$(d$,n)right$(r$,n+12)n;:go
to120
```


"BARGRAF VIC" Listing

by Eddie Johnson

```
1 rem **** program 3 ****
2 rem ***** bargraf *****
3 rem ***** vic  20 *****
100 input"[clr]increment (1-100)";i:ifi<
1ori>100then100
110 d$="[home]":fors=1to22:d$=d$+"[down]
":e$=e$+" ":l$=l$+"[left]":next:b$="[rvs
 on]"+e$:e$="[right][right]"+e$+l$
120 print"[clr][down][down][down]";:forc
=65to73:printchr$(c):print:next
130 input"[home][rvs on]which letter";l$
:l=asc(l$)*2-126:ifl<3orl>20then130
140 print"[home][down]number (1"i*-15"[l
eft])";:inputn:ifn>i*15orn<1then140
150 printleft$(d$,l)e$left$(b$,1+int(n/i
))"[rvs off]"n;:goto130
```


"BARGRAF C64" Listing

by Eddie Johnson

```
1 rem **** program 4 ****
2 rem ***** bargraf *****
3 rem *****  c-64  ******
100 print"[clr]maximum input(33-3300)":i
nputm:i=m/33:ifm<33orm>3300then100
110 d$="[home]":fors=1to40:d$=d$+"[down]
":e$=e$+" ":l$=l$+"[left]":next:b$="[rvs
 on]"+e$:e$="[right][right]"+e$+l$
120 print"[clr][down][down][down]";:forc
=65to74:printchr$(c):print:next
130 input"[home][rvs on]which letter";l$
:l=asc(l$)*2-126:ifl<3orl>22then130
140 print"[home][down]number (1"m*-1"[le
ft])";:inputn:ifn>morn<1then140
150 printleft$(d$,l)e$left$(b$,1+int(n/i
))"[rvs off]"n;:goto130
```

BASIC ALLEY:
Questions Answered FRE(Ø)

by Bob Richardson

Readers occasionally phone or write to us with programming questions, and nearly all of the questions turn out to be the same ones others have asked before. Each of us seems to plod through the learning process at his (her) own pace — but regardless of how fast we may learn, we still must tread the same path all others have taken before us. Only a handful of people are pioneers. The rest of us are relegated to following.

In keeping with that line of reasoning, it is obvious that there can never be a question that is too "old", or too "stupid", or too "elementary" to ask. We should never be embarassed to ask a question. After all, we should remember that there are others out there who don't know enough about the subject to be *able* to ask the questions.

So, it occurs to us that perhaps a column for questions and answers might be helpful to everyone concerned. Not only would it provide a place to ask your particular question, but it would also provide a place to read and learn about brand new questions — ones you may not otherwise have come across for some time to come. So send in or phone in your questions. We will attempt to answer them all, printing the more common ones as space will allow.

Here are a couple of the more frequently-asked questions to get the ball rolling:

Q — Can you explain the purpose of the FRE function and how to use it?

A — The FRE function tells you or your program how much memory is free for your BASIC program and variables. To use this function on all Commodore computers *except* the C-64, use one of the following formats:

```
10 X=FRE(0)
Or,
PRINT FRE(0)
```

The first example will find the amount of free memory and store it in the variable X. The PRINT statement in the next line will tell you free memory while you are programming. The number in parentheses can be any number, and has no effect on the outcome. However, you *must* have a number in parentheses because BASIC treats all functions in this manner.

On the Commodore 64, things get more complicated. First, some background: In the early days of the PET, 8032, and VIC-20, memory expansion did not exceed 32K. The FRE function stored its data in BINARY form, which is a series of zeros and ones. Suffice it to say that this number could not be very large; it had a range from −32768 to +32767. This was fine for the earlier computers because memory would almost never exceed 32K. On the C-64, though, if more than 32K is available, the number will "wrap around" to negative. The solution is rather simple: If free memory is less than zero, then add 65536 (that's 32768×2) to the negative number. (The C-128 handles FRE in a different way, and does not exhibit this problem.) This is achieved through the following statements:

```
10 X=FRE(0)-(FRE(0)<0)*65536
Or,
PRINT FRE(0)-(FRE(0)<0)*65536
```

One final note on the FRE function: If you are writing a program and use FRE after entering a line, the function will only tell you how much free space is left after your program, and will *not* include the space used by variables. Even if you have enough memory for additional program lines, there may not be enough room to RUN the program if you use lots of strings or large arrays.

Q — Why do BASIC programs that use integers seem to run slower (or, at least, no faster) than programs that use decimal numbers? Aren't integers easier for the computer?

A — You are correct that programs using integers are slower than those using floating point (decimal) numbers. To understand why, you must first understand the concept behind integers on Commodore computers.

Decimal numbers are stored in memory in a series of five bytes. If you have large arrays of numbers (such as three dimensional arrays), memory is consumed rapidly. That is why the people who designed Commodore BASIC (and many other BASICs, as well) installed the integer feature. Integers occupy only two bytes of memory, so you can store 2.5 times as much information in memory than you can when using floating point numbers (providing, of course, that your variables do not have decimal points in them). The time delay problem arises because BASIC must *convert* each integer into a floating point number before performing calculations, and then back to an integer again to store the result.

Why, then, you may well ask, with standard memory sizes growing by leaps and bounds, do we even fool with integers? One good reason is that if you *compile* your program using one of the popular BASIC compilers, the integer calculations *will* be performed faster, and you will still save memory space.

Those are our questions for this time. If you wish to see the answers to *your* questions appear here, a good way to proceed toward that end would be to write them down and send them in. Remember, the only dumb question is the one that goes unasked. We look forward to hearing from you.

COMAL —
How To Draw Almost Anything

by Valerie Jean Kramer

A friend of mine got me started some time back when he asked for help in programming a Keno game in COMAL. Thanks to that project, I got past the learning curve of COMAL and realized what a treasure I had in the language. Well, that same friend has been at it again. I suggested he try a simple project by hinself: a COMAL program to draw a "Kilroy" picture. You know, a brick wall with a simple face peering over the top. He tried, but I found that he really didn't have the first idea where to start. So, for those of you in the same boat, here is how you can draw Kilroy or anything else. I have no artistic talent worth mentioning, so I will leave details of composition to your imagination. I want to show what the various COMAL graphics statements are, how to use them, and how to produce the simple geometric patterns from which many pictures are made.

The COMAL graphic commands are shown in the list below, grouped by function. We will see many of them in this article. You will see many of them in this article. You may wish to look up those we don't use. Most of them are pretty simple once you have a general idea of what is going on and I hope you have that idea by the end of the article.

```
SETGRAPHIC      SETTEXT
FULLSCREEN      SPLITSCREEN
BACKGROUND      BORDER          PENCLOLR
HIDETURTLE      SHOWTURTLE      TURTLESIZE
PENDOWN         PENUP
DRAWTO          MOVETO          SETXY
LEFT            RIGHT           SETHEADING
BACK            FORWARD
HOME
CLEAR
FRAME
PLOT            FILL
PLOTTEXT
```

The first thing you need to do is to get into COMAL. From COMAL you must enter graphics mode:

```
SETGRAPHIC 0
```

That puts you in the high-res graphics mode. (There is also a multi-color mode, `SETGRAPHIC 1`, which works pretty much the same as high-res. Let's save it for another time.) At this point, you should see a blank screen with a triangular shaped turtle in the center and a couple of lines across the top of the screen in a different color. The large area is your graphic screen. The small area at the top is a text area where you can give commands. Let's try a couple of commands just for fun. Press the RETURN key after each command.

```
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
```

Did your turtle draw a box? You can clear your screen and start over with the command `CLEAR`. You can return the turtle to the starting position without clearing the screen by using the `HOME` command. Try them now.

```
HOME
CLEAR
```

We can control the turtle with the `HIDETURTLE`, `SHOWTURTLE`, AND `TURTLESIZE` commands. Try these:

```
HIDETURTLE
SHOWTURTLE
TURTLESIZE 0
TURTLESIZE 5
TURTLESIZE 10
```

Next, let's get a good map of the graphic screen in our mind. There are 64,000 distinct points on the screen. They are arranged in a rectangle 200 points high by 320 wide. Each point is labeled by two numbers. The first number tells how far the point is from the left edge and the second tells how far from the bottom of the screen. Thus the lower left point is 0,0 because it is at the left edge and on the bottom. The top right point is 319,199. The center of the screen is called the "home" position of the turtle. It is located at point 160,99. Here is a mock-up of a screen, showing some locations:

```
0,199
160,199
319,199
0,99
160,99
319,99
0,0
160,0
319,0

(x,y)
```

Now that we know how our points are numbered, we can turn on any point we want using the PLOT command. PLOT x,y turns on the single point located at the

specified x,y position. Let's try some and get a feel for our screen.

```
PLOT 160,200
PLOT 140,99
PLOT 140,95
PLOT 10,10
PLOT 315,195
PLOT 5,195
PLOT 317,5
```

Continue playing until you have a good feel for the screen coordinate system.

The next geometric figure is the straight line. From geometry, we know that two points determine a line. We need a starting and an ending point. To draw a line on our screen, we first move to the starting point of our line, and then draw a line to the ending point. There are a couple ways of doing this. Here are some examples:

```
SETXY 23,99
FORWARD 51

MOVETO 25,99
DRAWTO 25,150
```

Or:

```
SETXY 235,185
FORWARD 36

MOVETO 235,180
DRAWTO 270,180
```


How about lines at angles? Sure!

```
SETXY 170,150
SETHEADING 135
FORWARD 75

MOVETO 160,150
DRAWTO 210,100
```

Try some of your own. When you are done, clear the screen with the `CLEAR` command. Note that the `SETHEADING` command turns the turtle to the specified angle where zero derees is straight up, 90 degrees is right, etc. Some projects will be more easily drawn with `MOVETO` and `DRAWTO` while others will do better with `SETXY`, `SETHEADING` and `FORWARD` type commands.

Now, let's try for some polygons, starting with a triangle. We move to the first point, draw to the second, draw to the third, then draw to the first again.

```
MOVETO 140,99
DRAWTO 160,129
DRAWTO 180,99
DRAWTO 140,99
```

I'll leave other straight-lined figures for you to work out on your own. You know how to put any kind of a straight line on the screen at any place you wish, so you should really have no problems.

Since we are going to get into more complicated figures, it's time to change our method of operation. Instead of typing in each command and watching it execute, we will write a program containing the commands. When it is just the way we want it, we will RUN it and have our picture drawn for us. This has several advantages. We can save our program for repeated use. We can make use of loops and other program constructs that are not available as direct commands. When we make a goof, we need only correct the program. We don't have to type all the other commands over, too.

Let's get out of graphics mode. You can give the command `SETTEXT` or just hit the **f1** key. If you type `SETGRAPHIC` without a number, or hit the **f3** key you will be back in split screen graphics mode, and can review your picture. **f5** puts you in full screen graphics mode. You can also change modes with the `SPLITSCREEN` and `FULLSCREEN` commands but the function keys are simpler to use. Try playing with them now, then return to text mode(**f1**).

Since we start our program from text mode, the first thing it needs to do is to enter graphics mode. We will not be giving commands in graphics mode (our program does that for us) so let's use the fullscreen graphics. Finally let's hide the turtle so we can see only the points and lines we've drawn. Our program will begin like this:

```
10 // Comment to identify our program
20 setgraphic 0
30 fullscreen
40 hideturtle
```

I haven't said anything about colors yet, but let's set our background and border colors. We may need to look up the color numbers in the Commodore Programmer's Reference Guide. COMAL uses the same color numbers

used by BASIC. Black is 0, white is 1, red is 2, green is 5, blue is 6, etc.

```
50 background 1
60 border 5
70 clear
```

The CLEAR command will erase the graphics screen and set it all to the color we last specified with the BACKGROUND command. We can also set the color of the lines we want drawn. We do this with the PENCOLOR statement.

```
80 pencolor 6
```

We're all set now, but what shall we do? Let's draw the square we drew earlier in the command mode.

```
90 forward 50
100 right 90
110 forward 50
120 right 90
130 forward 50
140 right 90
150 forward 50
```

Now run it. Because we are in program mode, we can make use of loop and conditional structures. FOR-ENDFOR loops are particularly useful for polygons. Let's improve the square program with a FOR loop.

```
0010 // improved square program
0020 setgraphic 0
0030 fullscreen
0040 hideturtle
0050 background 1
0060 border 5
0070 clear
0080 pencolor 6
0090 for side = 1 to 4
0100  forward 50
0110  right 90
0120 endfor side
```

With a simple change, we can make this program print any polygon. Add line 15 and replace lines 90 thru 120 with this:

```
0015 input "how many sides? ":number

0090 //
0100 for sides = 1 to number do
0110  forward 10
0120  right 360/number
0130 endfor sides
```

Try some programs of your own. You might also want to use the LEFT and BACK commands. They work like RIGHT and FORWARD but go in the other direction.

In order to draw curved lines, we cheat. The COMAL statements necessary have already been written for us and are found in the *Library of Functions and Procedures* book/disk set. It includes procedures that will draw circles and arcs for us. All we have to do is merge the procedures from the library disk into our program and call them as needed. The book explains how to call each routine.

To merge a library procedure or function with your program, you do not use any line numbers greater than 8999. The library routines start at 9000. When you are ready, you ENTER the library routine. This will merge it from disk and put it at the end of your program. Remember that ENTER works only with SEQ files (as created by LISTing a program section to disk)!

The library disk is set up so ENTER will work. After the library routine is entered, renumber your program to make room for any additional routines you may want to merge:

```
RENUM
```

If you forget to renumber, the next routine you ENTER will overlay all or part of the first one and you will have a real mess.

It's time to start drawing Kilroy. Kilroy consists of a wall of bricks (let's keep it simple with only three rows of bricks) with Kilroy's head peeking over the wall. For extra credit, you ambitious ones can add Kilroy's fingers and nose or extend the picture in other ways. For now, we need to draw two objects — a brick wall, and a head. The brick wall can be sen in several ways:

1. A lot of distinct lines.
2. A box with two horizontal lines and a bunch of vertical ones.
3. Three rows of bricks.

The third alternative is probably the best, because it is the most general description. We can easily draw a brick. It's just a variation of our square program. Next, we can learn to draw a row of bricks and, finally, we can draw three such rows. By abstracting the program this way, we will later be able to change the size of the brick, the number of bricks in a row, the number of rows, etc. With the other two approaches, these changes would be much more difficult.

In fact, to keep things very modular, we will put each of our abstractions into its own procedure. Let's write the BRICK PROC:

```
310 proc brick(high,wide) closed
320  for side:-1 to 2 do
330   forward high
340   right 90
350   forward wide
360   right 90
370  endfor side
380  //
390  left 90
400  back wide
410  right 90
420 endproc brick
```

Notice that we had to use separate FORWARD commands for the two adjacent sides since they may not be the same length. The LEFT, BACK, RIGHT statements at the end of the proc move the turtle to the right end of the brick in position to draw an adjacent brick. This will make our procedure to draw a row of bricks very simple:

```
440 proc row'of'bricks(count,high,wide)
closed // wrap line
450  for i:=1 to count do
460   brick(high,wide)
470  endfor i
480 endproc row'of'bricks
```

There is only one small problem. Succeeding rows nof bricks alternate a short and a long brick at the start of the row. We can make a second proc for the other row of bricks.:

```
500 proc other'row'of'bricks(count,high,
wide) closed // wrap line
510  short:=int(wide/2)
520  brick(high,short)
530  //
540  for i:=1 to count-1 do
550   brick(high,wide)
560  endfor i
570  //
580  short:=wide-short
590  brick(high,short)
600 endproc other'row'of'bricks
```

We had to be careful in our computations because we don't want to end up with one row of bricks shorter than another. We are now ready for our wall procedure:


```
620 proc brick'wall(x,y,rows,count,high,
wide) closed // wrap line
630  moveto x,y
640  case (rows mod 2) of
650  when 0
660   for i:=1 to rows/2 do
670    row'of'bricks(count,high,wide)
680    moveto x,y-(2*i*high)
690    other'row'of'bricks(count,high,wi
de)
700    moveto x,y-((2*i+1)*high)
710   endfor i
720  when 1
730   row'of'bricks(count,high,wide)
740   moveto x,y-high
750   for i:=1 to (rows-1)/2 do
760    other'row'of'bricks(count,high,wi
de)
770    moveto x,y-(2*i*high)
780    row'of'bricks(count,high,wide)
790    moveto x,y-((2*i+1)*high)
800   endfor i
810  endcase
820 endproc brick'wall
```

Again, we had to be careful because we might call for either an even or odd number of bricks. Note that we must also be sure to set our position before drawing each row of bricks.

Our main program is very much like it used to be:

```
100 // for comal 0.14 - kilroy was here
110 // delete"0:c14.kilroy"
120 // save  "0:c14.kilroy"
130 setgraphic 0
140 fullscreen
150 hideturtle
160 background 1
170 border 5
180 clear
190 pencolor 0
200 brick'wall(20,80,3,10,10,25)
210 arc(160,90,40,0,180) // head
```

Finally, we start tinkering and add the bells and whistles. We add eyes and a nose as well as a title for the picture. Here is the final version as far as I have taken it. Now it's your turn to experiment both with this picture and others of your own creation. Good luck and have fun!

```
0100 // for comal 0.14 - kilroy was here
0110 // delete"0:c14.kilroy"
0120 // save  "0:c14.kilroy"
0130 setgraphic 0
0140 fullscreen
0150 hideturtle
0160 background 1
0170 border 5
0180 clear
0190 pencolor 0
0200 brick'wall(20,80,3,10,10,25)
0210 arc(160,90,40,0,180) // head
0220 pendown
0230 moveto 150,115
0240 brick(4,4) // eye
0250 moveto 170,115
0260 brick(4,4) // eye
0270 moveto 160,100
0280 brick(6,4) // mouth
0290 plottext 100,10,"kilroy was here"
0300 //
0310 proc brick(high,wide) closed
0320  for side:=1 to 2 do
```

```
0330    forward high
0340    right 90
0350    forward wide
0360    right 90
0370   endfor side
0380   //
0390   left 90
0400   back wide
0410   right 90
0420 endproc brick
0430 //
0440 proc row'of'bricks(count,high,wide)
 closed // wrap line
0450   for i:=1 to count do
0460    brick(high,wide)
0470   endfor i
0480 endproc row'of'bricks
0490 //
0500 proc other'row'of'bricks(count,high
,wide) closed // wrap line
0510   short:=int(wide/2)
0520   brick(high,short)
0530   //
0540   for i:=1 to count-1 do
0550    brick(high,wide)
0560   endfor i
0570   //
0580   short:=wide-short
0590   brick(high,short)
0600 endproc other'row'of'bricks
0610 //
0620 proc brick'wall(x,y,rows,count,high
,wide) closed // wrap line
0630   moveto x,y
0640   case (rows mod 2) of
0650   when 0
0660    for i:=1 to rows/2 do
0670     row'of'bricks(count,high,wide)
0680     moveto x,y-(2*i*high)
0690     other'row'of'bricks(count,high,w
ide) // wrap line
0700     moveto x,y-((2*i+1)*high)
0710    endfor i
0720   when 1
0730    row'of'bricks(count,high,wide)
0740    moveto x,y-high
0750    for i:=1 to (rows-1)/2 do
0760     other'row'of'bricks(count,high,w
ide) // wrap line
0770     moveto x,y-(2*i*high)
0780     row'of'bricks(count,high,wide)
0790     moveto x,y-((2*i+1)*high)
0800    endfor i
0810   endcase
0820 endproc brick'wall
0830 //
0840 proc arc(x,y,r,sa,ca) closed
0850   moveto x,y
0860   setheading 90-sa
0870   penup
0880   forward r
0890   left 90
0900   pendown
0910   arcl(r,ca)
0920   penup
0930 endproc arc
0940 //
0950 proc arcl(r,ca) closed
0960   for i:=1 to ca/10 do
0970    left 5
0980    forward r*3.14159/18
0990    left 5
1000   endfor i
1010 endproc arcl
```


■

128 Memory Map

Compiled by Chris Scott

This is the first part of a two-part article I'll be writing on the memory of the 128 PC in 128 mode. This part will include the map itself with a little bit on the use of the MMU (Memory Management Unit) chip. The last part will be a bit more visual because I will be illustrating what is where inside your 128.

Those of you who program exclusively in BASIC should be as interested in the memory map as the assembler programmers are. Even though BASIC sweats all the details for you, someday you may need to POKE or PEEK a number in memory. Knowing what the 128 does with your POKE, and how to use what you PEEK could be helpful.

Knowing the memory layout of your computer also enhances your capabilities in mixing machine language with your BASIC, so that you can speed up those routines that take the patience out of your personality.

The first area that should be of interest to you is in the beginning (lowest numbered) memory. Whether working with BASIC or not, there are some very useful locations in an area often referred to as the Zero Page. Memory can be divided into "pages" in much the same way that you might talk about "the twenty-four hundered block" location in a city — the term derives from the high order digits of the address. The high order digits of the Zero Page are, of course, zero, and the microprocessor in the 128 (and 64) have fast instructions that can only be used in this area. Programmers like to take advantage of these speedy instructions, so it is a very busy area of memory.

For instance, take location 215 ($00D7 in hexidecimal). In the 64 mode, it is the location to find the last ASCII character to be printed to the screen, but in the 128 mode it will tell you whether the 40- or 80-column display is being used. What could be more simple for your program when you need to see if you should have the user switch?

With this map you can find other golden tidbits of your computer's memory. In the next issue, I'll show in a glance what the MMU knows and keeps track of for you. With VIC owners, memory configurations are probably a matter of normal events, but few 64 owners had to worry about what configuration they were in. The final part will take some of the mystery out of the MMU and show how to use it to your best advantage. See you next time.

```
Configurations --

              0      - RAM Bank 0 only
              1      - RAM Bank 1 only
              2,3    - For Future Expansion, currently identical to 0,1
              4-7    - Internal Function ROM, RAM Banks 0-3 respectively
              8-11   - External Cartridge ROM, RAM Banks 0-3 respectively
              12     - Kernal and Internal ROM low, RAM Bank 0
              13     - Kernal and External ROM low, RAM bank 0
              14     - Kernal, BASIC, RAM Bank 0, and Character ROM
              15     - Kernal, BASIC, RAM Bank 0

NOTE -  Any bank containing ROM also contains I/O except Bank 14.  Also,
        BASIC defaults to Bank 15.  Bank 15 below $F4000 is best because
        it contains I/O.

        Examples - LDA #$0E:STA $FF00 - for RAM Bank 0
                   LDA #$4E:STA $FF00 - for RAM Bank 1

MMU Programming technique - LDA $FF00:PHA:
                            do your routine:
                            PLA:STA $FF00:RTS

================================================================================
                          ----   RAM LEVEL 0    ----

================================================================================
MEMORY                              DEFAULT          DESCRIPTIONS,
LOCATIONS                           BYTE/WORD        USAGE,
HEX (DECIMAL)     LABEL             HEX (DECI)       & PARAMETERS
================================================================================

0000 D510 (0)                     %00101111 (47)     8510 DATA DIRECTION REGISTER
0001 R510 (1)                     %01110011 (115)    8510 DATA REGISTER

0002-0089 (0-137)              BASIC ZERO PAGE
```

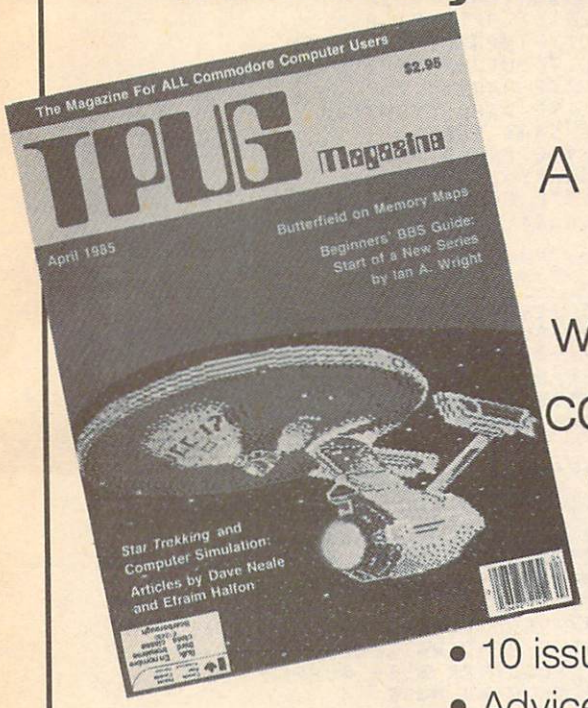
MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)		DESCRIPTIONS, USAGE, & PARAMETERS
0002-0009	(0-9)		TEMPORARY STORAGE		
0002	(2)	BANK	$0F	(15)	BANK
		PCB			MONITOR PC TEMP
0003	(3)	PCHI	$B0	(176)	>PC
		PCH			MONITOR PC TEMP
0004	(4)	PCLO	$00	(0)	<PC
		PCL			MONITOR PC TEMP
0005	(5)	SREG	$00	(0)	.ST
		FLGS			MONITOR STATUS TEMP
0006	(6)	AREG	$00	(0)	.A
		ACC			ACCUMULATOR TEMP
0007	(7)	XREG	$00	(0)	.X
		XR			X REGISTER TEMP
0008	(8)	YREG	$00	(0)	.Y
		YR			Y REGISTER TEMP
0009	(9)	STKPTR	$F8	(284)	STACK POINTER
		CHARAC			SEARCH CHARACTER P FROM CPU OR CODE FOR SWITCH TO 64 MODE
		SP			STACK POINTER TEMP
000A	(10)	ENDCHR	$00	(0)	FLAG- SCAN FOR QUOTE
000B	(11)	TRMPOS	$00	(0)	SCREEN COLUMN FROM LAST TAB
000C	(12)	VERCK	$00	(0)	FLAG- 0=LOAD 1=VERIFY
000D	(13)	COUNT	$FA	(250)	INPUT BUFFER POINTER/ NUMBER OF SUBSCRIPTS
000E	(14)	DIMFLAG	$00	(0)	FLAG- DEFAULT ARRAY DIMENSION
000F	(15)	VALTYP	$00	(0)	DATA TYPE- $FF (255)=STRING / $00 (0)=NUMERICAL
0010	(16)	INTFLG	$00	(0)	DATA TYPE- $80 (128)=INTEGER / $00 (0)=FLOATING POINT
0011	(17)	GARBFL	$00	(0)	FLAG- DATA SCAN / LIST QUOTE/ GARBAGE COLLECTION
		DORES			
0012	(18)	SUBFLG	$00	(0)	FLAG-SUBSCRIPT REFERENCE / USER FUNCTION CALL
0013	(19)	INPFLG	$00	(0)	FLAG- $00 (0)=INPUT (0) / $40 (64)=GET / $98 (152)=READ
0014	(20)	DOMASK	$00	(0)	
		TANSGN			FLAG-TAN SIGN/COMPARISON RESULT
0015	(21)	CHANNL	$00	(0)	CURRENT I/O PROMPT FLAG
		POKER	$00	(0)	INTEGER VALUE - (<LB)
0016-0017	(22-23)	LINNUM	$00	(0)	- (>HB)
0018	(24)	TEMPPT	$1B	(27)	TEMPORARY STRING STACK POINTER
0019	(25)	LASTPT	$00	(0)	<LAST TEMPORARY STRING ADDRESS
001A	(26)	LASTPT+1	$00	(0)	>LAST TEMPORARY STRING ADDRESS
001B	(27)	TEMPST	$00	(0)	STACK FOR TEMPORARY STRINGS
001C	(28)	TEMPST+1	$00	(0)	
001D	(29)	TEMPST+2	$00	(0)	
001E	(30)	TEMPST+3	$00	(0)	
001F	(31)	TEMPST+4	$00	(0)	
0020	(32)	TEMPST+5	$00	(0)	
0021	(33)	TEMPST+6	$00	(0)	
0022	(34)	TEMPST+7	$00	(0)	
0023	(35)	TEMPST+8	$00	(0)	
0024	(36)	INDEX	$06	(6)	<UTILITY POINTER AREA
0025	(37)	INDEX1	$02	(2)	>UTILITY POINTER AREA
0026	(38)	INDEX2	$00	(0)	
0027	(39)	INDEX2+1	$00	(0)	
0028	(40)	RESHO	$00	(0)	FLOATING POINT PRODUCT OF MULTIPLY
0029	(41)	RESMOH	$00	(0)	
002A	(42)	ADDEND	$00	(0)	
		RESMO			
002B	(43)	RESLO	$00	(0)	
002C	(44)		$00	(0)	
002D-003C	(45-60)		BASIC MEMORY VECTORS		
002D	(45)	TXTTAB	$01	(1)	<START OF BASIC PROGRAM - RAM LEVEL (0)
002E	(46)		$1C	(28)	>START OF BASIC PROGRAM - RAM LEVEL (0)
002F	(47)	VARTAB	$00	(0)	<START OF VARIABLES $0400 - RAM LEVEL (1)

MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)		DESCRIPTIONS, USAGE, & PARAMETERS
0030	(78)		$04	(4)	>START OF VARIABLES $0400 - RAM LEVEL (1)
0031	(49)	ARYTAB	$00	(0)	<START OF ARRAYS - RAM LEVEL (1)
0032	(50)		$04	(4)	>START OF ARRAYS - RAM LEVEL (1)
0033	(51)	STREND	$00	(0)	<END OF ARRAYS + 1 - RAM LEVEL (1)
0034	(52)		$04	(4)	>END OF ARRAYS + 1 - RAM LEVEL (1)
0035	(53)	FRETOP	$00	(0)	<BOTTOM OF STRINGS - RAM LEVEL (1)
0036	(54)		$FF	(255)	>BOTTOM OF STRINGS - RAM LEVEL (1)
0037	(55)	FRESPC	$00	(0)	<LOWEST STRING VECTOR - RAM LEVEL (1)
0038	(56)		$00	(0)	>LOWEST STRING VECTOR - RAM LEVEL (1)
0039	(57)	MXMEM1	$00	(0)	<TOP OF STRINGS - RAM LEVEL (1)
003A	(58)		$FF	(255)	>TOP OF STRINGS - RAM LEVEL (1)
003B	(59)	CURLIN	$00	(0)	<CURRENT BASIC LINE NUMBER
003C	(60)		$FF	(255)	>CURRENT BASIC LINE NUMBER
003D	(61)	TXTPTR	$01	(1)	<POINTER TO BASIC TEXT
003E	(62)		$02	(2)	>POINTER TO BASIC TEXT
003F	(63)	FORM	$00	(0)	USED BY PRINT USING
		FNDPNT			<POINTER TO ITEM FOUND BY SEARCH
0040	(64)		$00	(0)	>POINTER TO ITEM FOUND BY SEARCH
0041	(65)	DATLIN	$00	(0)	<CURRENT DATA LINE NUMBER
0042	(66)		$00	(0)	>CURRENT DATA LINE NUMBER
0043	(67)	DATPTR	$00	(0)	<CURRENT DATA ADDRESS
0044	(68)		$1C	(28)	>CURRENT DATA ADDRESS
0045	(69)	INPPTR	$00	(0)	<VECTOR-INPUT ROUTINE
0046	(70)		$00	(0)	>VECTOR-INPUT ROUTINE
0047	(71)	VARNAM	$00	(0)	<CURRENT BASIC VARIABLE NAME
0048	(72)		$00	(0)	>CURRENT BASIC VARIABLE NAME
0049	(73)	FDECPT	$00	(0)	
		VARPNT			<POINTER- CURRENT BASIC VARIABLE DATA
004A	(74)		$00	(0)	>POINTER- CURRENT BASIC VARIABLE DATA
004B	(75)	LSTPNT	$00	(0)	
		ANDMSK			
		FORPNT			<POINTER- INDEX VARIABLE FOR/NEXT
004C	(76)	EORMSK	$00	(0)	
		FORPNT+1			>POINTER- INDEX VARIABLE FOR/NEXT
004D	(77)	VARTXT	$00	(0)	
		OPPTR			
004E	(78)		$00	(0)	
004F	(79)	OPPMASK	$00	(0)	
0050	(80)	GRBPNT	$00	(0)	
		TEMPF3			
		DEFPNT			
0051	(81)		$00	(0)	
0052	(82)	DSCPNT	$00	(0)	
0053	(83)		$00	(0)	
0054	(84)		$00	(0)	
0055	(85)	HELPER	$00	(0)	FLAGS 'HELP' OR 'LIST'
0056	(86)	JMPER	$4C	(76)	
0057	(87)		$00	(0)	
0058	(88)	OLDOV	$00	(0)	USED TO DIM ARRAYS
0059	(89)	TEMPF1	$00	(0)	MULTIPLY DEFINITION FOR INSTRUCTION
		PTARG1			=TEMPORARY F1
005A	(90)	TEMPF1+1	$00	(0)	=TEMPORARY F1+1
		ARYPNT			
		HIGHDS			
005B	(91)	TEMPF1+2	$00	(0)	=TEMPORARY F1+2
		PTARG2			
005C	(92)	TEMPF1+3	$00	(0)	=TEMPORARY F1+3
		HIGHTR			
005D	(93)	STR1	$00	(0)	=TEMPORARY F1+4
005E	(94)	TEMPF2			=TEMPORARY F1+5
005F	(95)	DECCNT			=TEMPORARY F1+6
0060	(96)	STR2	$08	(8)	=TEMPORARY F1+7
		TØ			MONITOR ZP STORAGE IN FAC
0061	(97)	GRBTOP	$00	(0)	=TEMPORARY F1+8
		DPTFLG			
		LOWTR			
0062	(98)	EXPSGN	$00	(0)	=TEMPORARY F1+9
		TENEXP			=DECCNT+1

MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)		DESCRIPTIONS, USAGE, & PARAMETERS
0063	(99)	POSITN	$00	(0)	=TEMPORARY F1+10
		FAC			
		DSCTMP			
		LEFTFLAG			PAINT-LEFT FLAG
		FACEXP			FAC#1 EXPONET
		T1			MONITOR ZP STORAGE IN FAC
0064	(100)	MATCH	$00	(0)	=TEMPORARY F1+11
		RIGHTFLAG			PAINT-RIGHT FLAG
		FACHO			
0065	(101)	FACMOH	$00	(0)	
0066	(102)	INDICE	$66	(66)	
		FACMOH			
		FACMO			
		T2			MONITOR ZP STORAGE IN FAC
0067	(103)	FACLO	$00	(0)	
0068	(104)	FACSGN	$0F	(15)	POINTER- SERIES-EVAL. CONST.
0069	(105)	DEGREE	$00	(0)	
		SGNFLG			POINTER- SERIES-EVAL. CONST.
006A	(106)	ARGEXP	$00	(0)	FAC#2 EXPONET
006B	(107)	ARGHO	$00	(0)	FAC#2 MANTISSA
006C	(108)	ARGMOH	$00	(0)	
		INITASØ			JUST A COUNT FOR INIT
006D	(109)	ARGMO	$00	(0)	
006E	(110)	ARGLO	$00	(0)	
006F	(111)	ARGSGN	$00	(0)	FAC#2 SIGN
0070	(112)	STRNG1	$00	(0)	
		ARISGN			SIGN COMPARISON RESOLUTION FAC#1 VS #2
0071	(113)	FACOV	$00	(0)	FAC#1 LOW-ORDER (ROUNDING)
0072	(114)	STRNG2	$00	(0)	
		POLYPT			
		CURTOL			
		FBUFPT			<POINTER- CASSETTE BUFFER
0073	(115)	FBUFPT+1	$00	(0)	>POINTER- CASSETTE BUFFER
0074	(116)	AUTINC	$00	(0)	INCREMENT VALUE FOR AUTO (0=OFF)
0075	(117)		$00	(0)	
0076	(118)	MVDFLG	$00	(0)	FLAG IF 10K HIRES
0077	(119)	NOZE	$D0	(208)	USING LEAD ZERO COUNT
		SPRNUM			MOVSPR & SPRITE TEMPORARY
		KEYNUM			
		ZPTEMP1			
0078	(120)	HULP	$00	(0)	COUNTER
		KEYSIZ			
0079	(121)	SYNTMP	$00	(0)	TEMPORARY FOR INDIVIDUAL LOADS
007A	(122)	DSDESC	$07	(7)	DESCRIPTOR FOR DS$
		TXTPTR			
007B	(123)		$00	(0)	
007C	(124)		$00	(0)	
007D	(125)	TOS	$FF	(255)	TOP / RUNTIME STACK
007E	(126)		$09	(9)	
007F	(127)	RUNMOD	$00	(0)	FLAG- RUN/DIRECT MODE
0080	(128)	PARSTS	$00	(0)	DOS PARSER ST WORD
		POINT			USING POINTER/DECIMAL POINTER
0081	(129)	PARSTX	$00	(0)	
0082	(130)	OLDSTK	$FA	(250)	GRAPHIC ZP STORAGE
0083-008F	(131-143)		BASIC Z-P STORAGE FOR GRAPHIC COMMANDS		
0083	(131)	COLSEL	$00	(0)	CURRENT COLOR
0084	(132)	MULTIC1	$01	(1)	CURRENT MULTI-COLOR #1
0085	(133)	MULTIC2	$02	(2)	CURRENT MULTI-COLOR #2
0086	(134)	FG	$0D	(13)	CURRENT FOREGROUND COLOR
0087	(135)	SCALEX	$00	(0)	<SCALE FACTOR IN X
0088	(136)		$00	(0)	>SCALE FACTOR IN X
0089	(137)	SCALEY	$00	(0)	<SCALE FACTOR IN Y
008A	(138)		$00	(0)	>SCALE FACTOR IN Y
008B	(139)	STOPNB	$00	(0)	STOP PAINT IF NOT BACKGROUND OR SAME COLOR
008C	(140)	GRAPNT	$00	(0)	
008D	(141)	GRAPNT+1	$00	(0)	
008E	(142)	VTEMP1	$00	(0)	
008F	(143)	VTEMP2	$00	(0)	
0090-00FF	(144-255)		KERNAL/EDITOR STORAGE		
0090	(144)	STATUS	$00	(0)	I/O OPERATION STATUS BYTE

TPUG

MORE THAN JUST ANOTHER COMPUTER MAGAZINE!



A membership in the world's largest computer club will provide you with:

- 10 issues of TPUG magazine
- Advice from experts like Jim Butterfield and Elizabeth Deal
- Access to our huge library of public domain software
- Access to our online help services
- All for only $25/yr.

TPUG
101 Duncan Mill
Suite G7
Don Mills, Ontario
CANADA
M3B 1Z3

JOIN US NOW!!

I want to join TPUG as an associate member.

Name _____ Home phone (____) _____

Address _____ Work phone (____) _____

City/Town _____ ☐ Cheque enclosed

Prov/State _____ ☐ Money order enclosed

Postal/Zip Code _____ Country _____ ☐ Visa ☐ MasterCard

My computer equipment is: _____ Card# _____

_____ Expiry _____

_____ Signature _____

MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)		DESCRIPTIONS, USAGE, & PARAMETERS
0091	(145)	STKEY	$FF	(255)	STOPFLAG- STOP KEY
0092	(146)	SVXT	$80	(128)	TAPE TEMPORARY
0093	(147)	VERCK	$00	(0)	LOAD OR VERIFY FLAG
0094	(148)	C3P0	$55	(85)	SERIAL BUFFERED CHARACTER
0095	(149)	BSOUR	$FF	(255)	CHARACTER BUFFER FOR SERIAL
0096	(150)	SYNO	$00	(0)	CASSETTE SYNC BUFFER
0097	(151)	XSAV	$00	(0)	TEMPORARY FOR BASIN
0098	(152)	LDTND	$00	(0)	NUMBER OF OPEN FILES
0099	(153)	DFLTN	$00	(0)	DEFAULT INPUT DEVICE (0)
009A	(154)	DFLTO	$03	(3)	DEFAULT OUTPUT DEVICE (3)
009B	(155)	PRTY	$00	(0)	CASSETTE PARITY
009C	(156)	DPSW	$00	(0)	CASSETTE DIPOLE SWITCH
009D	(157)	MSGFLG	$C0	(192)	OS MESSAGE FLAG
009E	(158)	PTR1	$00	(0)	CASSETTE ERROR PASS 1
		T1			TEMPORARY 1
009F	(159)	PTR2	$00	(0)	CASSETTE ERROR PASS 2
		T2			TEMPORARY 2
00A0-00A2	(160-161)		JIFFY CLOCK		
00A0	(160)	TIME	$xx		65536 JIFFYS (18.2044 minutes)
00A1	(161)	TIMMNS	$xx		256 JIFFYS (4.2267 seconds)
00A2	(162)	TIMSEC	$xx		JIFFY (.01667 second)
00A3	(163)	R2D2	$55	(85)	SERIAL BUSS USAGE
		PCNTR			CASSETTE STUFF
00A4	(164)	BSOUR1	$00	(0)	TEMPORARY/SERIAL ROUTINE
		FIRT			
00A5	(165)	COUNT	$00	(0)	TEMPORARY/SERIAL ROUTINE
		CNTDN			CASSETTE SYNC COUNTDOWN
00A6	(166)	BUFPNT	$00	(0)	CASSETTE BUFFER POINTER
00A7	(167)	INBIT	$00	(0)	RS232 RECEIVER INTERNAL BIT
		SHCNL			CASSETTE SHORT COUNT
00A8	(168)	BITCI	$00	(0)	RS232 RECEIVER BIT COUNT IN
		RER			CASSETTE READ ERROR
00A9	(169)	RINONE	$00	(0)	RS-232 RECEIVER FL FOR START BIT CHECK
		REZ			CASSETTE READING ZEROS
00AA	(170)	RIDATA	$00	(0)	RS232 RECEIVER BYTE BUFFER
		RDFLG			CASSETTE READ MODE
00AB	(171)	RIPRTY	$00	(0)	RS232 RECEIVER PARITY
		SHCNH			CASSETTE SHORT COUNT
00AC	(172)	SAL	$00	(0)	POINTER - TAPE BUFFER/SCREEN SCROLL
00AD	(173)		$0B	(11)	
00AE	(174)	EAL	$00	(0)	TAPE END ADDRESSES/END OF PROGRAM
00AF	(175)		$00	(0)	
00B0	(176)	CMP0	$00	(0)	TAPE TIME CONST.
00B1	(177)	TEMP	$00	(0)	
00B2	(178)	TAPE1	$00	(0)	TAPE BUFFER VECTOR
00B3	(179)		$0B	(11)	
00B4	(180)	BITTS	$00	(0)	RS-232 TRANSFER BIT COUNTER
		SNSW1			
00B5	(181)	NXTBIT	$00	(0)	RS-232 TRANSFER NEXT BIT
		DIFF			
00B6	(182)	RODATA	$00	(0)	RS-232 TRANSFER BYTE BUFFER
		PRP			
00B7	(183)	FNLEN	$01	(1)	LENGTH OF CURRENT FILENAME
00B8	(184)	LA	$00	(0)	CURRENT FILE NUMBER
00B9	(184)	SA	$6F	(111)	CURRENT SECONDARY ADDRESS
00BA	(185)	FA	$08	(8)	CURRENT DEVICE NUMBER
00BB	(186)	FNADR	$16	(22)	<FILENAME ADDRESS
00BC	(187)		$FA	(250)	>FILENAME ADDRESS
00BD	(188)	ROPRTY	$00	(0)	RS-232 TRANSFER PARITY
		OCCHAR			OUTPUT CHARACTER
00BE	(190)	FSBLK	$00	(0)	# BLOCKS LEFT TO READ/WRITE
00BF	(191)	DRIVE	$30	(48)	
		MYCH			SERIAL WORD BUFFER
00C0	(192)	CAS1	$00	(0)	CASSETTE MANUAL/CONTROLLED SWITCH (IRQ)
00C1	(193)	TRACK	$01	(1)	
		STAL			<IO START ADDRESS
00C2	(194)	SECTOR	$00	(0)	
		STAH			>IO START ADDRESS
00C3	(195)	MEMUSS	$73	(115)	<CASSETTE LOAD TEMPORARYS
		TMP2			
00C4	(196)	MEMUS+1	$E0	(224)	>CASSETTE LOAD TEMPORARYS
		TMP2+1			

MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)		DESCRIPTIONS, USAGE, & PARAMETERS
00C5	(197)	DATA	$00	(0)	TAPE READ WRITE DATA
00C6	(198)	BA	$00	(0)	BANK FOR CURRENT LOAD/SAVE/VERIFY OPERATIONS
00C7	(199)	FNBANK	$0F	(15)	BANK OF FILENAME
00C8	(200)	RIBUF	$00	(0)	<RS-232 INPUT BUFFER POINTER
00C9	(201)		$0C	(12)	>RS-232 INPUT BUFFER POINTER
00CA	(202)	ROBUF	$00	(0)	<RS-232 OUTPUT BUFFER POINTER
00CB	(203)		$0D	(13)	>RS-232 OUTPUT BUFFER POINTER
00CC-00D9	(204-217)		GLOBAL SCREEN EDITOR VARIABLES		
00CC	(204)	KEYTAB	$80	(128)	<KEYSCAN TABLE POINTER
00CD	(205)		$FA	(250)	>KEYSCAN TABLE POINTER
00CE	(206)	IMPARM	$B4	(180)	<PRIMARY UTIL STRING POINTER
00CF	(207)		$B5	(181)	>PRIMARY UTIL STRING POINTER
00D0	(208)	NDX	$00	(0)	INDEX/KEYBOARD QUEUE
00D1	(209)	KYNDX	$00	(0)	PENDING FUNCTION KEY FLAG
00D2	(210)	KEYIDX	$35	(53)	INDEX TO PENDING FUNCTION KEY STRING
00D3	(211)	SHFLAG	$00	(0)	SHIFT KEY STATUS
00D4	(212)	SFDX	$58	(88)	CURRENT KEY INDEX
00D5	(213)	LSTX	$58	(88)	LAST KEY INDEX
00D6	(214)	CRSW	$00	(0)	(CR) INPUT FLAG
00D7	(215)	MODE	$80	(88)	40/80 COLUMN MODE
00D8	(216)	GRAPHM	$00	(0)	TEXT/GRAPHIC MODE
00D9	(217)	CHAREN	$00	(0)	RAM/ROM VIC CHARACTER FETCH FLAG (BIT 2)
00DA-00DF	(218-223)	KEYVAR	THESE LOCATIONS ARE SHARED BY SEVERAL EDITOR ROUTINES		
00DA	(218)	SEDSAL	$0B	(11)	POINTERS FOR MOVELINE
		KEYSIZ			PROGRAMMABLE KEY VARIABLES
		BITMSK			TEMPORARY/TAB & LINE WRAP
00DB	(219)	KEYLEN	$00	(0)	
		SAVER			
00DC	(220)	SEDEAL	$00	(0)	
		KEYNUM			
00DD	(221)	KEYNXT	$00	(0)	
00DE	(222)	SEDT1	$00	(0)	
		SAVPOS			
		KEYBNK			
00DF	(223)	SEDT2	$0F	(15)	
		KEYTMP			
00E0-00EF	(224-239)		LOCAL SCREEN EDITOR VARIABLES. THESE ARE SWAPPED OUT TO $0A40 (2624) WHEN 40/80 MODE CHANGES. POINTERS FOR SCREEN EDITOR		
00E0	(224)	PNT	$00	(0)	<POINTER TO CURRENT LINE (TEXT)
00E1	(225)		$05	(5)	>POINTER TO CURRENT LINE (TEXT)
00E2	(226)	USER	$50	(80)	<POINTER TO CURRENT LINE (ATTRIBUTE)
00E3	(227)		$0D	(13)	>POINTER TO CURRENT LINE (ATTRIBUTE)
00E4	(228)	SCBOT	$18	(24)	<WINDOW LOWER LIMIT
00E5	(229)		$00	(0)	>WINDOW LOWER LIMIT
00E6	(230)	SCTOP	$00	(0)	<WINDOW UPPER LIMIT
00E7	(231)		$4F	(79)	>WINDOW UPPER LIMIT
00E8	(232)	LSXP	$02	(2)	CURRENT INPUT COLUMN START
00E9	(233)	LSTP	$06	(0)	CURRENT INPUT LINE START
00EA	(234)	INDX	$14	(20)	CURRENT INPUT LINE END
00EB	(235)	TBLX	$06	(0)	CURRENT CURSOR LINE
00EC	(236)	PNTR	$0C	(12)	CURRENT CURSOR COLUMN
00ED	(237)	LINES	$18	(24)	MAX NUMBER OF SCREEN LINES
00EE	(238)	COLUMN	$4F	(79)	MAX NUMBER OF SCREEN COLUMNS
00EF	(239)	DATAX	$20	(32)	CURRENT CHARACTER TO PRINT
00F0	(240)	LSTCHR	$20	(32)	PREVIOUS CHAR PRINTED (ESC)
00F1	(241)	COLOR	$82	(130)	CURRENT ATTRIBUTE TO PRINT (DEFAULT FOREGROUND COLOR)
00F2	(242)	TCOLOR	$82	(130)	SAVED ATTRIBUTE TO PRINT ('INSERT' & 'DELETE')
00F3	(243)	RVS	$00	(0)	REVERSE MODE FLAG
00F4	(244)	QTSW	$00	(0)	QUOTE MODE FLAG
00F5	(245)	INSRT	$00	(0)	INSERT MODE FLAG
00F6	(246)	INSFLG	$00	(0)	AUTO-INSERT MODE FLAG
00F7	(247)	LOCKS	$00	(0)	DISABLES <C=><SHIFT>, <CTRL> S
00F8	(248)	SCROLL	$00	(0)	DISABLES SCREEN SCROLL, LINE LINKER
00F9	(249)	BEEPER	$00	(0)	DISABLES <CTRL> G

```
================================================================================
MEMORY                                  DEFAULT         DESCRIPTIONS,
LOCATIONS                               BYTE/WORD       USAGE,
HEX (DECIMAL)                           HEX (DECI)      & PARAMETERS
================================================================================

00FA-00FE (250-254)             FREE ZERO PAGE RESERVED FOR APPLICATIONS SOFTWARE
                                  $FA-FE (250-254)

00FA        (250)  FREZP1           $07     (7)
00FB        (251)  FREZP2           $00     (0)
00FC        (252)  FREZP3           $00     (0)
00FD        (253)  FREZP4           $00     (0)
00FE        (254)  FREZP5           $00     (0)
00FF        (254)  LOFBUF           $00     (0)

0100-01FF (256-511) FBUFFR      CPU STACK

0100-010F (256-271)             BASIC DOS INTERFACE VARIABLES
                                AREA TO BUILD FILENAME IN (16 BYTES)

0110-0148 (272-328)             BASIC DOS REGISTERS

0110        (272)  XCNT                             DOS LOOP COUNTER
0111        (273)  DOSF1L                           DOS FILENAME 1 LENGTH
0112        (274)  DOSDS1                           DOS DISK DRIVE 1
0113        (275)  DOSF1A                           <DOS FILENAME 1 ADDRESS
0114        (276)                                   >DOS FILENAME 1 ADDRESS
0115        (277)  DOSF2L                           DOS FILENAME 2 LENGTH
0116        (278)  DOSDS2                           DOS DISK DRIVE 2
0117        (279)  DOSF2A                           <DOS FILENAME 2 ADDRESS
0118        (280)                                   >DOS FILENAME 2 ADDRESS
0119        (281)  DOSOFL                           <BLOAD/BSAVE START ADDRESS
011A        (282)                                   >BLOAD/BSAVE START ADDRESS
011B        (283)  DOSOFH                           <BSAVE ENDING ADDRESS
011C        (284)                                   >BSAVE ENDING ADDRESS
011D        (285)  DOSLA                            DOS LOGICAL ADDRESS
011E        (286)  DOSFA                            DOS PHYSICAL ADDRESS
011F        (287)  DOSSA                            DOS SECONDARY ADDRESS
0120        (288)  DOSRCL                           DOS RECORD LENGTH
0121        (289)  DOSBNK                           DOS BANK #
0122        (290)  DOSDID                           <DOS DISK ID
0123        (291)                                   >DOS DISK ID
0124        (292)  DIDCHK                           DOS DISK ID FLAG

0125-02E3 (293-739)             SPACE USED BY PRINT USING

0125        (293)  BNR                              POINTER TO BEGINNING #
0126        (294)  ENR                              POINTER TO ENDING #
0127        (295)  DOLR                             DOLLAR FLAG
0128        (296)  FLAG                             COMMA FLAG
0129        (297)  SWE                              COUNTER
012A        (298)  USGN                             SIGN EXPONENT
012B        (299)  UEXP                             POINTER TO EXPONENT
012C        (300)  VN                               NUMBER OF DIGITS BEFORE DECIMAL
                                                      POINT
012D        (301)  CHSN                             JUSTIFY FLAG
012E        (302)  VF                               NUMBER OF POSITION BEFORE DECIMAL
                                                      POINT (FIELD)
012F        (303)  NF                               NUMBER OF POSITION AFTER DECIMAL
                                                      POINT (FIELD)
0130        (304)  POSP                             +/- FLAG (FIELD)
0131        (305)  FESP                             EXPONENT FLAG (FIELD)
0132        (306)  ETOF                             SWITCH
0133        (307)  CFORM                            CHARACTER COUNTER
0134        (308)  SNO                              SIGN NUMBER
0135        (309)  BLFD                             BLANK/STAR FLAG
0136        (310)  BEGFD                            POINTER TO END OF FIELD
0137        (311)  LFOR                             LENGTH OF FORMAT
0138        (310)  ENDFD                            POINTER TO END OF FIELD

0200-02A1 (512-673)             BASIC AND MONITOR INPUT BUFFER

0200        (512)  BUF                              BASIC INPUT BUFFER

02A2-02FB (674-763)             KERNAL RAM CODE

02A2        (674)  FETCH

02A2-02AE (674-686)             LDA ($ZP),Y FOR MMU=X
```

MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)	DESCRIPTIONS, USAGE, & PARAMETERS
02AA	(682)	FETVEC		ZERO-PAGE INDIRECT ADDRESS
02AF-02BD	(687-701)		STA ($ZP),Y FOR MMU=X	
02AF	(687)	STASH		
02B9	(697)	STAVEC		ZERO-PAGE INDIRECT ADDRESS
02BE-02CC	(702-716)		CMP ($ZP),Y FOR MMU=X	
02BE	(702)	CMPARE		ZERO-PAGE INDIRECT ADDRESS
02C8	(712)	CMPVEC		ZERO-PAGE INDIRECT ADDRESS
02CD-02CF	(712-719)		JSR $02E3 (739)	
02CD	(717)	JSRFAR		JSR XXXX TO ANY BANK & RETURN
02D0-02E2	(720-738)		STORE CPU REGISTERS IN $02-09 (2-9)	
02E3-02FB	(739-763)		RTI BASED ON $02-09 (2-9)	
02E3	(739)	JMPFAR		JMP XXXX TO ANY BANK
02FC-033B	(764-827)		VECTORS	
02FC	(764)	ESCFNVEC		VECTOR FOR ADDITIONAL FUNCTION ROUTINES
02FD	(765)			
02FE	(766)	BNKVEC		VECTOR FOR FUNCTION ROUTINES CARTRIDGE USERS
02FF	(767)			
0300	(768)	IERROR	$3F (63)	PRINT BASIC ERROR MESSAGE 4D3F (19775)
0301	(769)		$4D (77)	
0302	(770)	IMAIN	$C6 (198)	BASIC WARM START 4DC6 (19910)
0303	(771)		$4D (77)	
0304	(772)	ICRNCH	$0D (13)	TO CRUNCH (TOKENIZATION) 430D (17165)
0305	(773)		$43 (67)	
0306	(774)	IQPLOP	$51 (81)	BASIC TEXT LIST 5151 (20817)
0307	(775)		$51 (81)	(CHARACTER LIST)
0308	(776)	IGONE	$A2 (162)	BASIC CHARACTER DISPATCH 4AA2 (19106)
0309	(777)		$4A (74)	
030A	(778)	IEVAL	$DA (218)	BASIC TOKEN EVALUATION 78DA (30938)
030B	(779)		$78 (120)	
030C	(780)	IESCLK	$21 (33)	ESCAPE-TOKEN CRUNCH, 4321 (17185)
030D	(781)		$43 (67)	
030E	(782)	IESCPR	$CD (205)	...LIST, 51CD (20941)
030F	(783)		$51 (81)	
0310	(784)	IESCEX	$A9 (169)	...AND EXECUTE. 4BA9 (19369)
0311	(785)		$4B (75)	
0312	(786)	FREEVEC	$FF (255)	NOT USED
0313	(787)		$00 (0)	
0314	(788)	IIRQ/CINV	$65 (101)	IRQ HARDWARE INTERRUPT FA65 (64101)
0315	(789)		$FA (250)	
0316	(790)	IBRK/CBINV	$03 (3)	BRK INTERRUPT B003 (45059)
0317	(791)		$B0 (176)	
0318	(792)	INMI	$40 (64)	NMI NON-MASKABL INTERRUPT FA40 (64064)
0319	(793)		$FA (250)	
031A	(794)	IOPEN	$BD (189)	OPEN EFBD (61373)
031B	(795)		$EF (239)	
031C	(796)	ICLOSE	$88 (136)	CLOSE F188 (61832)
031D	(797)		$F1 (241)	
031E	(798)	ICHKIN	$06 (6)	CHKIN F106 (61702)
031F	(799)		$F1 (241)	
0320	(800)	ICHKOUT	$4C (76)	CHKOUT F14C (61772)
0321	(801)		$F1 (241)	
0322	(802)	ICLRCH	$26 (38)	CLRCHN F226 (61990)
0323	(803)		$F2 (242)	
0324	(804)	IBASIN	$06 (6)	CHRIN EF06 (61190)
0325	(805)		$EF (239)	
0326	(806)	IBSOUT	$79 (121)	CHROUT EF79 (61305)
0327	(807)		$EF (239)	
0328	(808)	ISTOP	$6E (110)	STOP F66E (63086)
0329	(809)		$F6 (246)	

MEMORY LOCATIONS HEX (DECIMAL)			DEFAULT BYTE/WORD HEX (DECI)		DESCRIPTIONS, USAGE, & PARAMETERS	
032A	(810)	IGETIN	$EB (235)		GETIN	EEEB (61163)
032B	(811)		$EE (238)			
032C	(812)	ICLALL	$22 (34)		CLALL	F222 (61986)
032D	(813)		$F2 (242)			
032E	(814)	EXMON	$06 (6)		MONITOR COMMAND VECTOR	B006 (45062)
032F	(815)		$B0 (176)			
0330	(816)	ILOAD	$6C (108)		LOAD	F26C (62060)
0331	(817)		$F2 (242)			
0332	(818)	ISAVE	$4E (78)		SAVE	F54E (62798)
0333	(819)		$F5 (245)			
0334-0348	(820-840)		EDITOR INDIRECT VECTORS			
0334	(820)	CTLVAC	$B9 (185)		EDITOR PRINT'CNTRL'IND	C7B9 (51129)
0335	(821)		$C7 (199)			
0336	(822)	SHFVAC	$05 (5)		EDITOR PRINT'SHIFTED'IND	C805 (51205)
0337	(823)		$A8 (168)			
0338	(824)	ESCVAC	$C1 (193)		EDITOR PRINT'ESCAPE'IND	C1C9 (49609)
0339	(825)		$C9 (201)			
033A	(826)	KEYVAC	$E1 (225)		EDITOR KEYSCAN LOGIC IND	C5E1 (50657)
033B	(827)		$C5 (197)			
033C-037F	(828-895)				KERNAL TABLES	
0380-03FF	(896-1023)				BASIC RAM CODE	
0380-039E	(896-926)				BASIC CHRGET ROUTINES	
039F-03D1	(927-977)				MISCELLANEOUS LDA ROUTINES	
0400-04FF	(1024-1279)				VIC TEXT SCREEN (VM NUMBER1)	
0800-09FF	(2048-2559)				BASIC RUN-TIME STACK	
0A00-0AFF	(2560-2815)				MONITOR & KERNAL ABSOLUTE VARIABLES	
0A00-0A01	(2560-2561)				BASIC COLD START 4000/3	
0A02	(2562)				RAM INITIALIZE IF =$A5 (165)	
0A04	(2563)				BASIC INITIALIZE IF BIT 0 SET	
0A05-0A1F	(2564-2591)					
0A20-0A7F	(2592-2687)				GLOBAL ABSOLUTE SCREEN DECREED	
0A80-0AB4	(2688-2740)				MONITORS DOMAIN	
0AC0	(2752)				NUMBER OF INTERNAL/EXTERNAL ROM's	
0AC1-0AC4	(2753-2756)				ACTIVE ROM FLAGS	
0B00-0BBF	(2816-3007)				CASSETTE BUFFER	
0BC0-0BFF	(3008-3071)				(DISK BOOT PAGE)	
0C00-0CFF	(3072-3327)				RS-232 INPUT BUFFER	
0D00-0DFF	(3328-3583)				RS-232 OUTPUT BUFFER	
0E00-0FFF	(3584-4095)				SPRITE DEFINITION AREA	
1000-10FF	(4096-4351)				FUNCTION KEY BUFFER	
1000-1009	(4096-4105)				NUMBER OF BYTES PER KEY	
100A-10FF	(4107-4351)				FUNCTION KEY STRINGS	
1100-1107	(4352-4359)				CP/M RESET CODE	
1108-11FF	(4360-4607)				BASIC DOS/VSP VARIABLES	
1200-12FF	(4608-4863)				BASIC ABSOLUTE VARIABLES	
117A-122A	(4474-4650)				MISCELLANEOUS VECTORS	
1210	(4624)				END OF BASIC PROGRAM	
1300-13FF	(4864-5119)					
1400-17FF	(5120-6143)				RESERVED/FOREIGN LANGUAGE SYSTEM	
1800-1BFF	(6144-7167)				RESERVED/FUNCTION KEY ML	
1C00-FEFF	(7168-65279)				BASIC PROGRAM AREA OR	
1C00-1FFF	(7168-8191)				VIC BIT-MAP COLOR (VM NUMBER2)	
2000-3FFF	(8192-16383)				VIC BIT-MAP SCREEN	
4000-FEFF	(16384-65279)				BASIC PROGRAM AREA (CONTINUED)	
FF05-FF44	(65285-65348)				KERNAL DISPATCH CODE	
FFD0-FFF9	(65488-65529)				CP/M and KERNAL RAM CODE	
FFFA-FFFB	(65530-65531)				NMI VECTOR	
FFFC-FFFD	(65532-65533)				RESET VECTOR	
FFFE-FFFF	(65534-65535)				IRQ VECTOR	

--== RAM LEVEL 1 ==--

0000-03FF	(0-1023)	COMMON WITH RAM LEVEL (0)
0400-FEFF	(1024-65279)	BASIC VARIABLE STORAGE
FF05-FF44	(65285-65348)	KERNAL INTERNAL DISPATCH CODE
FFF5-FFF9	(65525-65529)	"cbm" AND RESET ADDRESS $E224 (57892)
FFFA-FFFB	(65530-65531)	NMI VECTOR
FFFC-FFFD	(65532-65533)	RESET VECTOR
FFFE-FFFF	(65534-65535)	IRQ VECTOR

```
================================================================================
MEMORY                              DEFAULT          DESCRIPTIONS,
LOCATIONS                           BYTE/WORD        USAGE,
HEX (DECIMAL)                       HEX (DECI)       & PARAMETERS
================================================================================

================================================================================
                              --==  ROM-I/O  ==--
================================================================================

0000*    (0)                                        8502 ON-CHIP D-D REGISTERS
0001*    (1)                                        8502 ON-CHIP I/O REGISTERS


================================================================================
                              --==  BASIC ROM  ==--
================================================================================
FFFE-FFFF (65534-65535)                             IRQ VECTOR
4000-7FFF (16384-32767)                             BASIC LOW ROM OR
                                                      INTERNAL LOW ROM OR
                                                      EXTERNAL LOW ROM.
4000-4002 (16384-16386)                             BASIC POWER-UP JMP $4023 (16419)
4003-4022 (16387-16418)                             BASIC RESET    JMP $4009 (16393)
4023-4044 (16419-16452)                             BASIC POWER-UP
4045-410F (16453-16655)                             INITIALIZE BASIC REGISTERS
4112-4179 (16658-16761)                             INITIALIZE BASIC ABSOLUTE VARIABLES
417A-418C (16762-16780)                             INITIALIZE $D501-4 (54529-54532)
419B-41BA (16795-16826)                             PRINT RESET SCREEN
41BB-4250 (16827-16976)                             RESET SCREEN HEADER CHARACTER
4251-4278 (16977-17016)                             INITIALIZE $0300-11 (768-785) $02FC (764)
4279-43DD (17017-17373)                             BASIC CHRGET ML
430D-4416 (17165-17430)                             TOKENIZE BASIC TEXT
4417-4515 (17431-17685)                             BASIC 2.0 KEYWORDS
4516-46F8 (17686-18168)                             BASIC 7.0 KEYWORDS
484B-4A81 (18507-19073)                             BASIC ERROR MESSAGES
4AA2-4D3E (19106-19774)                             BASIC CHARACTER DISPATCH
4D3F-4DC5 (19775-19909)                             PRINT BASIC MESSAGE
4DC6-5150 (19910-20816)                             BASIC WARM START
5151-51D8 (20817-20952)                             BASIC TEXT LIST
51D9-5261 (20953-21089)                             INITIALIZE BASIC PROGRAM VECTORS
6EB2-6EDA (28338-28338)                             INITIALIZE BASIC ABSOLUTE VARIABLES
78DA-7E81 (30938-32385)                             BASIC TOKEN EVALUATION
7E82-7FFD (32386-32765)                             BLANK


================================================================================
                           --==  BASIC MIDDLE ROM  ==--
================================================================================

8000-BFFF (32768-49151)                             BASIC MIDDLE ROM OR INTERNAL ROM
                                                      OR EXTERNAL ROM
9251-9298 (37457-37528)                             BASIC TO KERNAL JMP TABLES
A845-A84C (43077-43084)                             TO BANK(15)
AA6E-AE62 (43630-44642)                             BLANK
AF00-AFA7 (44800-44967)                             JMP TABLES
AFA8-AFFF (44968-45055)                             BLANK
B000-BFFF (45056-49151)                             MONITOR ROM
B000-     (45056-)                                  MONITOR RESET CODE
B003-     (45059-)                                  BRK ENTRY
BB72-BFFD (47986-49149)                             BLANK


================================================================================
                              --==  ROM EDITOR  ==--
================================================================================
C000-CFFF (49152-53247)                             EDITOR HIGH ROM OR
                                                      INTERNAL ROM OR
                                                      EXTERNAL ROM
C000-     (49152-)                                  CINT -JMP $C07B (49275)
C00F-     (49167-)                                  SCREEN
C012-     (49170-)                                  SCNKEY
C018-     (49176-)                                  PLOT
C07B-     (49275-)                                  CINT
CEA8-CEF4 (52904-52980)                             FUNCTION KEY INITIALIZE VALUES
CEF5-CFFD (52981-53245)                             BLANK
```

```
================================================================================
MEMORY                              DEFAULT         DESCRIPTIONS,
LOCATIONS                           BYTE/WORD       USAGE,
HEX (DECIMAL)                       HEX (DECI)      & PARAMETERS
================================================================================

================================================================================
                          ---==  GRAPHIC/MMU REGISTERS  ==--
================================================================================

D000-DFFF (53248-57343)                             I/O SPACE
D500-D50B (54528-54539)                             MMU I/O CHIP (SWITCHABLE)
D500      (54528)                                   MMU CONFIGURATION REGISTER $FF00 (65280)
D501-D504 (54529-54532)                             MMU PRECONFIGURATION REGISTRATIONS
D505      (54533)                                   BIT  7   - 40/80 SWITCH 1=40
                                                         6   - 128/64 MODE  1=64
                                                         5   - GAME LINE       BI-
                                                                               DIRECTIONAL
                                                         4   - EXROM LINE      BI-
                                                                               DIRECTIONAL
                                                         3   - FAST SERIAL I/O
                                                         1-2 - RESERVED
                                                         0   - 8502/Z80 CPU 0=Z80
D506      (54534)                                   RAM CONFIGURATION REGISTRATION
                                                    BITS 0-1 -K OF COMMON RAM
                                                             00-1K
                                                             01-4K
                                                             10-8K
                                                             11-16K
                                                    BITS 2-3 -LOCATION/COMMON RAM
                                                             00-NONE
                                                             01-BOTTOM
                                                             10-TOP
                                                             11-BOTH
                                                    BITS 4-5 -RESERVED
                                                    BIT  6   -RAM NUMBER USED BY VIC
                                                    BIT  7   -RESERVED
D507      (54535)         PAGE FOR CPU ZERO PAGE
D508      (54536)                                   LSB-RAM NUMBER FOR CPU ZERO PAGE
D509      (54537)                                   PAGE FOR CPU STACK
D50A      (54538)                                   LSB-RAM NUMBER FOR CPU STACK
D50B      (54539)                                   SYSTEM VERSION REGISTERS
                                                    BITS 0-3 -MMU CHIP VERSIONS
                                                    BITS 4-7 -CODE FOR K/RAM
                                                             0010- 128K
                                                             0000- 256K
D50C-D5FF (54540-54783)                             BLANK
D600-D601 (54784-54785)                             ACCESS TO 16K DISPALY RAM
D600      (54784)                                   REGISTER ADDRESSES
D601      (54785)                                   DATA
DC00-DCFF (56320-56575)                             CIA #1 (KEYBOARD, ETC.)
DD00-DDFF (56576-56831)                             CIA #2 (SERIAL, ETC.)

================================================================================
                              --==  ROM KERNAL  ==--
================================================================================

E000-FFFF (57344-65535)                             KERNAL HIGH ROM
E000-     (57344-)                                  KERNAL ROM CODE
E000-E048 (57344-57416)                             RESET CODE
E04B-E055 (57419-57429)                             $D500 INITIALIZE VALUES
E056-     (57430-)                                  RESTORE
E05B-     (57435-)                                  VECTOR
E093-     (57491-)                                  RAMTAS
E0CD-E108 (57549-57608)                             INITIALIZE $FF05- ON RAM LEVEL (0-3) &
                                                      KERNAL RAM CODE ON RAM LEVEL (0)
E109-E1EF (57609-57839)                             I/O INITIALIZE ROUTINE
E1F0-E223 (57840-57891)                             IF $FFF5-9 ON RAM LEVEL (1)=cbm, THEN
                                                      JMP ($FFF8)- ELSE...
E224-E241 (57892-57921)                             INITIALIZE $FFF5-9 ON RAM LEVEL (1)
E242-E2BF (57922-58047)                             SWITCH TO 64 IF D505 (54533) BITS
                                                      4-5 NOT SET THEN CHECK INTERNAL/
                                                      EXTERNAL ROM AND JMP
E24B-E26A (57931-57962)                             SWITCH TO 64 MODE
E33B-     (58171-)                                  TALK
E33E-     (58174-)                                  LISTEN
E43E-     (58430-)                                  ACPTR
E4D2-     (58578-)                                  SECOND
E4E0-     (58592-)                                  TKSA
```

The Guide TO COMPUTER LIVING

A Monthly Publication For Commodore™ Owners

Formerly The Northwest Users Guide

Commodore support with a twist ... Personable and even humorous ... Timely news ... Helpful tutorials ... On-going support for several languages: BASIC (including BASIC 7.0 as featured in the new 128 PC), Machine Language, COMAL, Pascal ... Program Listings ... *Honest* Software reviews ... and much more ...

The Guide features some of the best computer humorists to be found.

- Introduce your "widow" to the Computer Widow's Compendium.
- Tutorials and feature articles by the infamous Mindy Skelton.
- Featuring Shelly Roberts' "I'm Sorry, But I Don't Speak Hexidecimal." Discover why Shelly just may be the Andy Rooney of the computer world!

We feel we have assembled one of the most talented staff of writers in the Commodore world. Receive each month the most friendly and helpful Commodore publication available. Written by Commodore users who are writing to you, not down at you.

Limited offer — While Supplies Last

Receive FREE High Quality Software with Each Subscription

One Year.....$18.00
(Canadian Rate–$24.00 U.S.*)

Receive FREE Either:

Omiterm Terminal Program

Fully supports the new 1660 Modem 300!
- Modem controls accessible from the keyboard
- Punter protocol — upload & download — 300/1200 baud
- Auto dial/re-dial
- Half/full duplex

$19.95 value.

OR:

Gold Disk

Choose one of the popular Canadian series disks — Each disk contains:
- Programming Tutorials
- Games
- Utilities

Top Quality Software!

$14.95 value.

Two Year.....$35.00
(Canadian Rate–$47.00 U.S.*)

Receive FREE Either:

Donald Duck's Playground

CES Software Showcase Award Winner! — Disney animation at its best! Children play four games to "earn" money to buy playground equipment. Builds money handling skills. Superb graphics. A bestseller!

$29.95 value.

OR:

Winnie The Pooh In The Hundred Acre Wood

Players explore the Hundred Acre Wood to find lost articles like Owl's books, Pooh's honey pot and Eeyore's tail, and return them to their rightful owners. Cheery music from the Disney movie caps off this computer rendition of the beloved classic.

$29.95 value.

Three Year.....$48.00
(Canadian Rate–$64.00 U.S.*)

Receive FREE:

Mind Prober

A personality program analysis program that helps you understand what makes your friends "tick". Answering a few simple questions about someone can reveal personality traits you might never have suspected. Amazingly accurate.

29.95 value

*** Canadian Orders Add $2.00 to Cover Software Shipping**

The Guide
3808 S.E. Licyntra Ct.
Portland, OR 97222
To Order by Phone, Call: (503) 654-5603

Name _____

Address _____

City, State & Zip _____

Please Enter My Subscription for:

☐ 1 Year ☐ 2 Years ☐ 3 Years

Please Check One: ☐ New ☐ Renewal

☐ VISA ☐ MasterCard

Card # _____

Exp. Date _____

Signature _____

My choice of free software, in accordance with the above offer, is:

_____

_____

```
================================================================================
MEMORY                              DEFAULT         DESCRIPTIONS,
LOCATIONS                           BYTE/WORD       USAGE,
HEX (DECIMAL)                       HEX (DECI)      & PARAMETERS
================================================================================
E503-      (58627-)                                 CIOUT
E515-      (58645-)                                 UNTLK
E526-      (58662-)                                 UNLSN
F23D-F264 (62013-62052)                             CLOSE ALL FILES ON DEVICE NUMBER-
                                                      ACCUMULATOR =DEVICE NUMBER
F265-      (62053-)                                 LOAD
F53E-      (62782-)                                 SAVE
F5F8-      (62968-)                                 UDTIM
F63D-F65D (63037-63069)                             CHECK FOR CTRL R/S OR C=
F65E-      (63070-)                                 RDTIM
F665-      (63077-)                                 SETTIM
F6B1-F71D (63153-63261)                             KERNAL MESSAGES
F731-      (63281-)                                 SETNAM
F738-      (63288-)                                 SETLFS
F744-      (63300-)                                 READST
F75C-      (63324-)                                 SETMSG
F75F-      (63327-)                                 SETTMO
F763-      (63331-)                                 MEMTOP
F772-      (63346-)                                 MEMBOT
F781-      (63361-)                                 IOBASE
F7D0-F7D9 (63440-63449)                             CALL KERNAL RAM CODE $02A2 (674)
                                                      TO LDA BYTE FROM ADDRESS,Y AT
                                                      ZERO-PAGE VECTOR IN ACCUMULATOR
                                                      FOR BANK(X)
F7EC-F7EF (63468-63471)                             LDA MMU VALUE FOR BANK(X)
F7F0-F7FF (63472-63487)                             MMU VALUES FOR BANK(0-15)
F800-F859 (63488-63577)                             KERNAL RAM CODE $02A2-02FB (674-763)
F85A-F866 (63578-63880)                             KERNAL RAM CODE $03F0-03FC (1008-1020)
F867-F988 (63591-63880)                             CHECK FOR ROM NOT USED ON RESET-
                                                      LOAD BOOT PAGE AND EXECUTE
F9D5-F9FA (63957-63994)                             LOAD PAGE AND STA  $ACC (2764)
FA00-      (64000-)                                 EDITOR TABLES
FA17-FA3F (64023-64063)                             PRINT BYTES AFTER JSR
FA40-      (64064-)                                 NMI
FA65-      (64101-)                                 IRQ
FC3B-FC7F (64571-64639)                             BLANK
FC80-FEFF (64640-65279)                             FOREIGN LANGUAGE SYSTEM-BLANK
FF00*      (65280)                                  MMU CONFIGURATION REGISTER
FF01-FF04*(65281-65284)                             MMU LOAD CONFIGURATION REGISTERS
FF05-FF44 (65285-65348)                             KERNAL INTERNAL DISPATCH CODE
FF3D-FF44 (65341-65348)                             RESET CODE JMP $E000 (57344)
FF47-FF80 (65351-65408)                             KERNAL HARDWARE JMP TABLE
FF81-FFF5 (65409-65525)                             KERNAL USER JUMP TABLE
FFFA-FFFB (65530-65531)                             NMI VECTOR
FFFC-FFFD (65532-65533)                             RESET VECTOR
FFFE-FFFF (65534-65535)                             IRQ VECTOR

================================================================================
  --==  THESE ARE I/O REGISTERS AND TAKE THE PLACE OF RAM OR ROM ALWAYS.  ==--
================================================================================

      MMU CONFIGURATION REGISTER $D500/FF00 (54528/65280)
       BITS VALUES PURPOSE
        6-7  RAM control-
             00     RAM LEVEL (0)
             01     RAM LEVEL (1)
             10     RAM LEVEL (2)-NOT PRESENT YET
             11     RAM LEVEL (3)-NOT PRESENT YET
        4-5  ROM CONTROL-
             00     KERNAL AND EDITOR ROM
             01     INTERNAL HIGH ROM
             10     EXTERNAL HIGH ROM
             11     NONE
        2-3  MIDDLE ROM CONTROL- $8000-BFFF (32768-36863)
             00     BASIC ROM
             01     INTERNAL ROM
             10     EXTERNAL ROM
             11     NONE
        1    LOW ROM CONTROL- $4000-7FFF (16384-32767)
             0      =ROM
             1      =RAM
        0    I/O-ROM CONTROL- $D000-DFFF (53248-57343)
             0      =I/O
             1      =CHARACTER ROM

================================================================================
```

User Group Directory

Alaska

Alaska 64 Computer Club
P.O. Box 6043
Anchorage, AK 99502

First City User's Group
P.O. Box 6692
Ketchikan, AK 99901
(907) 225-5695

Compooh-T
P.O. Box 118
Old Harbor, AK 99643
(907) 286-2213

Sitka Commodore User Group
P.O. Box 2204
Sitka, AK 99835

Alabama

Riverchase Commodore
User's Group
617 Grove Street
Birmingham, AL 35209
(205) 988-1078

Wiregrass Micro Computer
Society
109 Key Bent Road
Enterprise, AL 36330
(205) 347-7564

Huntsville PET User's Club
9002 Berclair Road
Huntsville, AL 35802

Commodore Club of Mobile
3868-H Rue Maison
Mobile, AL 36608
(205) 343-1178

Shoals Commodore User's
Group
209 Lakeshore Drive
Muscle Shoals, AL 35661

Tiger Byte Alabama CBM 64
Midway Plaza
Opelika, AL 36801

CC & ME
P.O. Box 324
Pinson, AL 35126
(205) 854-0650

Arizona

Central Arizona PET People
842 W. Calle Del Norte
Chandler, AZ 85224

Arizona VIC and 64 Users
904 W. Marlboro Circle
Chandler, AZ 85224
(602) 963-6149

Four Corners User's Group —
Canyon De Chelly
P.O. Box 1945
Chinle, AZ 86503
(602) 674-3421

VIC Users Group
2612 E. Covina
Mesa, AZ 85203

Arizona VIC-20/64 User's Club
232 W. 9th Place North
Mesa, AZ 85201

West Mesa VIC
2351 S. Standage
Mesa, AZ 85202

ACUG
c/o Home Computer Services
2028 West Camelback Road
Phoenix, AZ 85015
(602) 249-1186

Prescott Area Commodore
Club
P.O. Box 26532
Prescott Valley, AZ 86312

Thunder Mtn. Commodore
Computer
User's Group
P.O. Box 1796
Sierra Vista, AZ 85636

Catalina Commodore
Computer Club
Macey B. Taylor, Editor
3145 East Lee Street
Tucson, AZ 85716

Commodore User Group
c/o Metro Computer Store
4500 E. Speedway, Suite 13
Tucson, AZ 85712

Arkansas

Booneville 64 Club
401 W. 5th Street
Booneville, AR 72927

Commodore PET User's Club
Conway Middle School
Davis Street
Conway, AR 72032

Commodore Computer Club
P.O. Box 6000
South Station
Ft. Smith, AR 72906

VIC Club
c/o Hatfield Public School
P.O. Box 130
Hatfield, AR 71945
(501) 389-6164

River City Commodore Club
P.O. Box 4298
North Little Rock, AR 72116

Russellville Commodore User
Group
401 South Arlington Drive
Russellville, AR 72801
(501) 967-1868

Siloam Commodore Computer
Club
P.O. Box 88
Siloam Springs, AR 72761
(501) 524-5624

California

California Area Commodore
Terminal Users Society
P.O. Box 1277
Alta Loma, CA 91701

Pasadena Commodore
Computer Club
P.O. Box 1163
Arcadia, CA 91006

Auburn Commodore Club
Pat Strub
11210 Mira Loma Drive
Auburn, CA 95603

64 American Program
Exchange
3820 Brave Avenue
Bakersfield, CA 93309

Cal-Poly Commodore User's
Group
14617 1/2 Ramona Blvd.
Baldwin Park, CA 91706

VIC-20 Software Exchange
7660 Western Avenue
Buena Park, CA 90620

Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA 91506

Software 64
353 California Drive
Burlingame, CA 94010
(415) 340-7115

VACUUM
277 East 10th Avenue
Chico, CA 95926
(916) 891-8085

Fresno Commodore User's
Group
91 West 9th #203
Clovis, CA 93612

Amateurs and Artesians
Computing
P.O. Box 682
Cobb, CA 95426

CUTG
P.O. Box 1497
Costa Mesa, CA 92628
(714) 731-5195

Commodore 64 West User
Group
P.O. Box 346
Culver City, CA 90230

PUG of Silicon Valley
22355 Rancho Ventura Road
Cupertino, CA 95014

Valley Computer Club
P.O. Box 310
Denair, CA 95316

Southern California PET
User's Group
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361

Sphinx
7615 Leviston Avenue
El Cerrito, CA 94530
(415) 527-9286

Humboldt Commodore Users
Group
P.O. Box 6502
Eureka, CA 95501

Fairfield VIC-20 Club
1336 McKinley Street
Fairfield, CA 94533
(707) 427-0143

Sixty Forum
P.O. Box 16098
Fresno, CA 93755

VIC-20 Software Exchange
Club
10530 Sky Circle
Grass Valley, CA 95945

Diamond Bar Rop User's Club
2644 Amelgado
Hacienda Heights, CA 91745
(213) 333-2645

Commodore 64 Club
1804 North Dundee
Highland, CA 92346

Orange County VIC/C-64
User's Group
5832 Raphael Drive
Huntington Beach, CA 92649

C-64 West Orange County
User's Group
P.O. Box 1457
Huntington Beach, CA 92647
(714) 842-4484

Antelope Valley Commodore
User's Group
P.O. Box 4436
Lancaster, CA 93539
(805) 942-2626

Pals Livermore Society
886 South K
Livermore, CA 94550

The Exchange
P.O. Box 9189
Long Beach, CA 90810
(213) 595-1771

SLO VIC 20/64 Computer Club
Gary Bissell
1766 9th Street
Los Osos, CA 93402
(805) 528-3371

Manteca VIC-20 User's
Organization
429 N. Main Street
Manteca, CA 95336

Lincoln Computer Club
750 Yosemite
Manteca, CA 95336

Napa Valley Commodore
Computer Club
c/o Liberty Computerware
2680 Jefferson Street
Napa, CA 94558

Sacramento Commodore
User's Group
8120 Sundance Drive
Orangevale, CA 95662
(916) 969-2028

C-64/VIC 20 User's Group
Pasadena City College
Pasadena, CA 91106
(714) 593-4880

SIG
1135 Coronet Avenue
Pasadena, CA 91107

Success
301 Veronica Drive
Paso Robles, CA 93446

Commodore Owners of
Petaluma
877 Grant Avenue
Petaluma, CA 94952
(707) 762-8398

Diablo Valley Commodore
User's Group
762 Ruth Drive
Pleasant Hill, CA 94523
(415) 671-0145

Jurupa Wizards
8700 Galena Street
Riverside, CA 92509

Southern California Edison
Commodore Club
P.O. Box 800
Rosemead, CA 91770

Sacramento Commodore
Computer Club
P.O. Box 2227
Sacramento, CA 95810

Computer Barn Computer
Club
319 Main Street, Suite 2
Salinas, CA 93901

Peninsula Commodore User's
Group
549 Old County Road
San Carlos, CA 94070
(415) 593-7697

San Diego Commodore Users
Group
P.O. Box 86531
San Diego, CA 92138-6531

San Diego PUG
3562 Union Street
San Diego, CA 92103

S.D. East County C-64 User
Group
6353 Lake Apopka Place
San Diego, CA 92119
(619) 698-7814

S.F. Commodore Users Group
278 27th Avenue #103
San Francisco, CA 94121

VIC Club of San Francisco
1503 A Dolores
San Francisco, CA 94110

PET On The Air
525 Crestlake Drive
San Francisco, CA 94132

20/64
P.O. Box 18473
San Jose, CA 95158

Logiks Commodore Computer
Club
620 Del Ganado Road
San Rafael, CA 94903
(415) 479-0426

So. Bay Commodore 64
User's Group
P.O. Box 3193
San Ysindro, CA 95073

Commodore User's Group of
Santa Cruz
P.O. Box 8068
Santa Cruz, CA 95061-8068
(408) 335-2082

The Commodore Connection
2301 Mission Street
Santa Cruz, CA 95060
(408) 425-8054

Commodore User's Group
4237 Plumeria Court
Santa Maria, CA 93455

Commodore User's Club
1041 Foxenwoods Drive
Santa Maria, CA 93455

C-64 West
Harris Oliff
P.O. Box 406
Santa Monica, CA
90406-0406

VIC-20 User's Group
2791 McBride Lane — #121
Santa Rosa, CA 95407

Santa Rosa Commodore 64
User's Group
333 East Robles Avenue
Santa Rosa, CA 95407
(707) 584-7009

San Fernando Valley
Commodore
User's Group
1120 Whitecliff Road
Thousand Oaks, CA 91360

South Bay Commodore
User's Group
1402 W. 218th Street
Torrance, CA 90501

Commodore Interest
Association
c/o Computer Data
14660 La Paz Drive
Victorville, CA 92392

Walnut Creek PET User's Club
1815 Ygnacio Valley Road
Walnut Creek, CA 94596

Bay Area Computer
Association
1332 Pine Street
Walnut Creek, CA 94598
(415) 932-5447

Colorado

Rocky Mountain Commodore
Club
Ray Brooks
P.O. Box 377
Aspen, CO 81612

Aurora Market User's Group
15200 East 6th Avenue
Aurora, CO 80011

Colorado Commodore
Computer Club
Royce Mitchell
9400 Elm Court #6078
Denver, CO 80221

Colorado Commodore Club
2187 South Golden Court
Denver, CO 80227

Western Slope Commodore
Users Group
Ralph D'Andrea
P.O. Box 4142
Grand Junction, CO 81502

VICKIMPET User's Group
4 Waring Lane
Greenwood Village
Littleton, CO 80121

VICDORE User's Group
326 Emery Drive
Longmont, CO 80501
(303) 772-2821

Delaware

Newark Users Group
Robert Taylor
210 Durso Dr.
Newark, DE 19711

Florida

CHIPS
UMR Box 3063
Avon Park, FL 33825

Brandon User's Group
813 Valley Hill Drive
Brandon, FL 33511

Commodore Brooksville
User's Group
P.O. Box 1261
Brooksville, FL 33512
(904) 799-5292

Clearwater Commodore Club
1532 Lemon Street
Clearwater, FL 33516
(813) 442-0770

EL SHIFT OH
P.O. Box 548
Cocoa, FL 32922

RAM ROM 84
1620 Morning Dove Lane
Englewood, FL 33533
(813) 474-9450

Gainesville Commodore User
Group
Santa Fe Community College
Gainesville, FL 32602

Gainesville Commodore Users
3604-20A S.W. 31st Drive
Gainesville, FL 32608

Citrus Commodore User's
Group
P.O. Box 1494
Inverness, FL 32651
(904) 344-2793

Commodore Computer Club
P.O. Box 9726
Jacksonville, FL 32208

Jacksonville Area PET Society
401 Monument Road #177
Jacksonville, FL 32211

VIC/64 Heartland User's
Group
1220 Barrow Road #23
Lakeland, FL 33801
(813) 666-2132

Miami 20/64
12911 S.W. 49th Street
Miami, FL 33175
(305) 226-1185

The Ultimate 64 Experience
5740 S.W. 56th Terrace
Miami, FL 33143

64 Educators User's Group
South
9220 S.W. 52nd Terrace
Miami, FL 33165
(305) 274-3501

PETS and Friends
129 N.E. 44th Street
Miami, FL 33137

Richard Pristien
6278 S.W. 14th Street
Miami, FL 33144

64 User's Group
P.O. Box 561689
Miami, FL 33156

Commodore 64/VIC 20 User
Group
P.O. Box 5837, MP 142
Orlando, FL 32855

VIC User's Club
4071 Edgewater Drive
Orlando, FL 32804

Suncoast 64's
2395 US 19 North
Palm Harbor, FL 33563
(813) 785-1036

Bay Commodore User's Group
c/o Gulf Coast Computer Exch
241 N Tyndall Pkwy, Box 6215
Panama City, FL 32401

Commodore User's Group of Pensacola
P.O. Box 3533
Pensacola, FL 32516
(904) 455-5804

The Commodore Advantage
P.O. Box 18490
Pensacola, FL 32523
(904) 456-6554

Ft. Walton Beach Commodore User's Group
P.O. Box 3
Shalimar, FL 32579
(904) 651-3737

Commodore Computer Club
P.O. Box 21138
St. Petersburg, FL 33742

Commodore User's Group
545 E. Park Ave., Apt. 2
Tallahassee, FL 32301

Broward Commodore User's Group
P.O. Box 25794
Tamarac, FL 33320

Tampa Bay Commodore Computer Club
10208 North 30th Street
Tampa, FL 33612

South Florida PET User's Group
7170 S.W. 11th
West Hollywood, FL 33023
(305) 987-6982

Georgia

VIC Educators User's Group
c/o Cherokee County Schools
110 Academy Street
Canton, GA 30114

Commodore Computer Club of Columbus
6618 Foxdale Drive
Columbus, GA 31907
(404) 563-0828

Atlanta Commodore 64 User's Group
1767 Big Valley Lane
Stone Mountain, GA 30083
(404) 981-4253

Hawaii

20/64
P.O. Box 966
Kailua, HI 96734

Idaho

64 BUG
P.O. Box 276
Boise, ID 83701
(208) 344-6302

Commodore — Coeur D'Alene Computer Club
506 Lunceford Lane
Coeur D'Alene, ID 83814
(208) 765-3803

GHS Computer Club
Grangeville High School
910 South D Street
Grangeville, ID 83530

Eagle Rock Users Group
900 South Emerson
Idaho Falls, ID 83401

Commodore User's Group
310 Emerald Drive
Kellogg, ID 83837
(208) 784-8751

Pocatello Commodore Users Group
82 Mountain Drive
Pocatello, ID 83204

Commodore Users
548 East Center
Pocatello, ID 83201

SRHS Computer Club
Salmon River High School
Riggins, ID 83549

User's Group of Lower Idaho
Route 4, Box 67
Rupert, ID 83350
(208) 436-4283

Caribou Commodore Club
P.O. Box 535
Soda Springs, ID 83276
(208) 547-3921
(208) 547-4143

Illinois

East Side Computer Club
3103 Clay Street
Alton, IL 62002
(618) 462-7136

Gateway Computer Club
P.O. Box 207
Belleville, IL 62222

Canton Area Commodore User's Group
c/o Spoon River College
RR #1
Canton, IL 61520
(309) 647-4645 Ext. 255

Commodore 64 User's Club
104 Susan Lane
Carterville, IL 62918
(618) 985-4710

Champaign — Urbana Commodore User's Group
2006 Crescent Drive
Champaign, IL 61821
(217) 352-9681

Chicago Commodore 64 Users
P.O. Box 14233
Chicago, IL 60614

Commodore SIG Cache
Box C-176
323 Franklin, #804
Chicago, IL 60606
(312) 685-0994

Shelly Wernikoff
2731 N. Milwaukee Avenue
Chicago, IL 60647

McHenry County Commodore Club
227 East Terra Cotta Avenue
Crystal Lake, IL 60014
(815) 455-3942

Fox Valley Commodore Users Group
Herb Gross, Secretary
833 Prospect
Elgin, IL 60120
(312) 695-1316

Comcoe
2108 Sherman Avenue
Evanston, IL 60201

C-64 User's Group
P.O. Box 46464
Lincolnwood, IL 60646
(312) 588-0334

Survivors of Sixty-Four User's Group
WESL Institute
Western Illinois University
Macomb, IL 61455
(309) 298-2106
(309) 837-5378

Mt. Vernon Commodore Users
P.O. Box 512
Mt. Vernon, IL 62864

Central Illinois PET Users
635 Maple
Mt. Zion, IL 62549
(217) 864-5320

Fox Valley 64 User's Group
P.O. Box 28
N. Aurora IL 60542
(312) 898-2779

Oak Lawn Commodore Users
c/o The Computer Store
11004 South Cicero Avenue
Oak Lawn, IL 60453

Commodore User's Club
1707 East Main Street
Olney, IL 62450

VIC-20/64 User's Support
114 South Clark Street
Pana, IL 62557
(217) 562-4568

Peoria Area PET User's Group
800 S.W. Jefferson
Peoria, IL 61605
(309) 673-6635
(309) 674-5998

Illinois Valley Commodore User's Group
2330 12th Street
Peru, IL 61354
(815) 223-5141

Rockford Area PET Users
1608 Benton Street
Rockford, IL 61107

Springfield PET User Group
3116 Concord
Springfield, IL 62704
(217) 753-8500

Kankakee Hackers
RR 1, Box 279
St. Anne, IL 60964
(815) 933-4407

Sauk Valley Computer Club
P.O. Box 702
Sterling, IL 61081

Indiana

VIC/64 User's Group
2401 Columbus Avenue
Anderson, IN 46014
(317) 378-3016

Columbus Commodore Club
2676 Lafayette Avenue
Columbus, IN 47201

Commodore Computer Club
3814 Terra Terrace
Evansville, IN 47711
(812) 477-0739

Computer Workshop
VIC-20/64
282 South 600 West
Hebron, IN 46341
(219) 988-4535

PET/64 Users
10136 East 96th Street
Indianapolis, IN 46256

Commodore User's Group
1020 Michigan Avenue
Logansport, IN 46947
(219) 722-5205

Tri-State Commodore Users
6500 Center Ridge Road
Newburgh, IN 47630
(812) 853-2334

CHUG
12104 Meadow Lane
Oaklandon, IN 46236

Seymour Peekers
c/o D&L Camera Shop
108 North Chestnut
Seymour, IN 47274

Northern Indiana Commodore Enthusiasts
927 S. 26th Street
South Bend, IN 46615

Commodore 64 User's Group
912 S. Brown Avenue
Terre Haute, IN 47803

Commodore Owners of Lafayette
20 Patrick Lane
W. Lafayette, IN 47906
(317) 743-3410

Iowa

Commodore User Group
114 8th Street
Ames, IA 50010

Commo-Hawk Commodore User's Group
P.O. Box 2724
Cedar Rapids, IA 52406

Quad Cities Commodore Computer Club
P.O. Box 3994
Davenport, IA 52808
(319) 242-1496

Crawford County Commodore User Group
519 North 19th Street
Dennison, IA 51442
(712) 263-6274

Commodore Computer Users of Iowa
P.O. Box 3140
Des Moines, IA 50702

Commodore User's Group
965 Second Street
Marion, IA 52302

Newton Commodore User's Group
320 W. 9th Street S.
Newton, IA 50208
(515) 792-0814

Siouxland Commodore Club
2700 Sheridan Street
Sioux City, IA 51104
(712) 258-7903

Kansas

Walnut Valley Commodore User Group
1003 South Second Street
Arkansas City, KS 67005

Salt City Commodore Club
P.O. Box 2644
Hutchinson, KS 67501

Kansas Commodore Computer Club
101 South Burch
Olathe, KS 66061

Commodore User's Group
6050 South 183 Street West
Viola, KS 67149

Wichita Area PET Users
2231 Bullinnger
Wichita, KS 67204

Kentucky

C*BUG
P.O. Box 165
Bardstown, KY 40004
(502) 348-6380

Bowling Green Commodore User's Group
Rt. 11, Creekside Apt. 6
Bowling Green, KY 42101
(502) 781-9098

Louisville Users of Commodore KY
P.O. Box 22244
Louisville, KY 40222

Louisiana

64 Club
5200 Corporate Blvd.
Baton Rouge, LA 70808
(504) 925-5870

Commodore 64 User's Group
P.O. Box 1422
Baton Rouge, LA 70821

Commodore PET User Group
616 N. Niagara Circle
Gretna, LA 70053
(504) 394-4928

NOVA
917 Gordon Street
New Orleans, LA 70117
(504) 948-7643

Ark-La-Tex Commodore 64 Club
198 India Drive
Shreveport, LA 71115
(318) 797-9702

Commodore User's Group of Oachita
P.O. Box 175
Swaric, LA 71281
(318) 343-8044

Franklin Parish Computer Club
#3 Fair Avenue
Winnisboro, LA 71295

Maine

COM-VICS
RFD #1, Box 2086
Hebron, ME 04238
(207) 966-3641

SO. ME 64
10 Walker Street
Portland, ME 04102
(207) 761-1626

Compumania
81 North Street
Saco, ME 04072
(207) 282-7418

Northwoods Commodore User's Group
740 Main Street
Van Buren, ME 04785

Coastal Commodore Club
Waldoboro Village, Apt. 9
Waldoboro, ME 04572

Your Commodore User's Group
P.O. Box 611
Westbrook, ME 04092
(207) 854-4579

Maryland

Compucats
Betty Scheuler
680 W. Bel Air Ave.
Aberdeen, MD 20001

FOCUS
Gary Stone
P.O. Box 153
Annapolis, MD 20701

Viclique
Pat Foley
105A Conduit Street
Annapolis, MD 21001

Baltimore Commodore Users Group
John Reynolds
907 W. 36th Street
Baltimore, MD 21211

Westinghouse Commodore Users Group
Lee Barron
Mail Stop 5320 — P.O. Box 1693
Baltimore, MD 21203

Woodlawn Commodore Club
George Towner
1712 Aberdeen Road
Baltimore, MD 21234

The Boyds Connection
Tom DeReggi
21000 Clarksburg Road
Boyds, MD 20841

Southern Md. Commodore Users Group
Tom Helmke
6800 Kilarney Street
Clinton, MD 20735

Hyattsville C-64 Users Group
Kay Carpenter
7209 Dartmouth Ave.
College Park, MD 20740

Gaithersburg C-64 User Group
Russell Jarosinski
12937 Pickering Drive
Germantown, MD 20874

Jumpers 20/64's
Walt Marhefka
7837 B&A Ave.
Glen Burnie, MD 21061

Hagerstown User Group
Joe Rutkowski
23 Coventry Lane
Hagerstown, MD 21740

West Montgomery County C-64 User Group
Mark Richardson
8700 Hidden Hill Lane
Potomac, MD 20854

BAYCUG
Robert Smith
110 Danbury Road
Reisterstown, MD 21136

Rockville VIC/64 User Group
Tom Pounds
5112 Parklawn Terrace
Rockville, MD 20852

Montgomery County Commodore Users Group
Meryle Pounds
P.O. Box 6444
Silver Spring, MD 20906

Edison Commodore Users Group
Bill Harr
4314 Oxford Drive
Suitland, MD 20746

Massachusetts

Boston Computer Society — CBM Section
Jim Yost
577 Broadway
Boston, MA 02127

Eastern Mass 20/64
Frank Ordway
6 Flagg Road
Marlboro, MA 01752

Michigan

Downriver Commodore Group
17029 Keppen
Allen Park, MI 48101
(313) 274-2589

Tri-Cities User's Group
P.O. Box 45
Bay City, MI 48706
(517) 893-6999

South Computer Club
South Junior High School
45201 Owen
Belleville, MI 48111

Commodore User's Group
3947 W. Twelve Mile Road
Berkley, MI 48072

DEBUG
P.O. Box 196
Berrien Springs, MI 49103
(616) 471-1882

Quorum of 64
214 W. State Street
Clare, MI 48617
(517) 386-2251

Delton Area User's Group
11386 Letches Lane
Delton, MI 49046

David Liem
14361 Warwick Street
Detroit, MI 48223

Michigan's Commodore 64 User's Group
P.O. Box 539
E. Detroit, MI 48021
(313) 773-6302

Commodore User's Group
1010 Spicerville Hwy.
Eaton Rapids, MI 48827

Southeast Michigan PET Users
P.O. Box 214
Farmington, MI 48024

W. Michigan VIC-20/64 Users
1311 Portland, N.E.
Grand Rapids, MI 49505

Jackson Commodore Computer Club
201 South Grinnell Street
Jackson, MI 49203

Tri-County Computer Club
501 East Loomis
Ludington, MI 49431

COMP
486 Michigan Avenue
Marysville, MI 48040
(313) 364-6804

Commodore Computer Club
H. Dow High School--Rm. #226
Midland, MI 48640
(517) 835-5130

Edwardsburg Commodore User's Group
406 Brush Road
Niles, MI 49120
(616) 663-2792

SEM 64
2501 Five Mile, #3
Redford, MI 48239
(313) 537-4163

Traverse Area Commodore Club
(TACC)
David Shutler, Vice-President
200 High Lake Road
Traverse City, MI 49684
(616) 941-1179

VIC/64/PET User's Group
8439 Arlis Road
Union Lake, MI 48085

Commodore User Club
32303 Columbus Drive
Warren, MI 48093

DAB Computer Club
P.O. Box 542
Watervliet, MI 49098
(616) 463-5457

W. Michigan Commodores
1952 Cleveland Avenue S.W.
Wyoming, MI 49509
(616) 458-9724

Minnesota

MUPET
P.O. Box 179
Annandale, MN 55302

Brainerd Area Commodore User's Group
1219 S.E. 11th Street
Brainerd, MN 56401

Lake Superior Commodore
1936 Lawn Street
Duluth, MN 55812
(218) 728-3224

Heartland Area Computer Cooperative
Rt. 4, Box 204
Little Falls, MN 56345
(612) 632-5511

SMCUG
1002 Pfau Street
Mankato, MN 56001
(507) 625-6942

Twin Cities Commodore Club
6623 Ives Lane
Maple Grove, MN 55369
(612) 424-2425

Metro Area Commodore Club
Box M
Mendota, MN 55150

Club 64
256 16th Street, N.E.
Owatonna, MN 55060
(501) 451-0128

Mississippi

Gulf Coast Commodore Club
Mark W. Harvey
4550 West Beach Blvd.
Biloxi, MS 39531

Commodore Computer Club
So. Station, Box 10076
Hattiesburg, MS 39401

Commodore Biloxi User Group
3002 Hwy 90 East
Ocean Springs, MS 39564

Missouri

Heartland Users Group
Mason Emerson
129 S. Lorimer #7
Cape Girardeau, MO 64701

Mid-Missouri Commodore Club
1804 Vandiver Drive
Columbia, MO 65201

Commodore User Club of the Ozarks
211 North Aurora
Eldon, MO 65026

MOARK Commodore Users Group
Marshall B. Turner
P.O. Box 504
Golden, MO 65658

Joplin Commodore Computer Users
R. D. Connely
422 S. Florida Avenue
Joplin, MO 64801

KCPUG
5214 Blue Ridge Blvd.
Kansas City, MO 64133

Commodore Users Group of Kansas City
Salvador Cerda
P.O. Box 36492
Kansas City, MO 64111

PET Set Club of St. Louis
633 Bent Oak Drive
Lake St. Louis, MO 63367

Commodore PAC
Patricia Lucido
Horace Mann Learning Center
Maryville, MO 64468

Commodore User's Group of St. Louis
P.O. Box 6653
St. Louis, MO 63125-0653

Commodore Users Group of Warrensburg
Buck Sommerkamp
P.O. Box 893
Warrensburg, MO 64093
(816) 747-2406

Montana

Powder River Computer Club
Powder River High School
Broadus, MT 59317

Commodore User Club
1109 West Broadway
Butte, MT 59701

Western Montana Commodore User Group
Carl White
800 Kensington Avenue
Missoula, MT 59801

Nebraska

Alliance Commodore Computer Club
M. Seller
1629 Boise
Alliance, NE 69301

Pathfinder Commodore User's Group
1812 North I Street
Fremont, NE 68025

Platte Valley Commodore Users Group
Jim Parks
1720 'O' Street
Gering, NE 69341

Computer Power Unlimited
Robert Howard
416 North 27th
Lincoln, NE 68503
(402) 475-8081

Greater Omaha C64 Users Group
Ken Jirele / Bob Quisenberry
2932 Leawood Drive
Omaha, NE 68123
(402) 292-2753

Nevada

Las Vegas Area PET Users Group
5130 East Charleston Blvd.
Suite 5-315
Las Vegas, NV 89122

SOG Commodore Users Group
Lee Schram
4011 West Charleston Blvd.
Las Vegas, NV 89102

Silver State Commodore User's Group
P.O. Box 81075
Las Vegas, NV 89180

Compu Club 64
4220 S. Maryland Parkway
Bldg. B, Suite 403
Las Vegas, NV 89109
(702) 369-7354

C-Run
Franklin Miller
P.O. Box 70473
Reno, NV 89570

New Hampshire

Northern N.E. Computer Society
P.O. Box 69
Berlin, NH 03570

C-64 User's User Software Exchange
P.O. Box 4022
Rochester, NH 03867

Monadnock Commodore 64 Users
RFD #1, Route 10
Winchester, NH 03470

New Jersey

South Jersey Commodore Users Group
Mark Orthner
P.O. Box 420
Cherry Hill, NJ 08034

Somerset Users Group
Stan Kosciuk
5 Shasta Pass
Fanwood, NJ 07023

RCA Commodore Users Group
William Rowe
432 Hemlock Lane
Mount Holly, NJ 08060

New Mexico

New Mexico Commodore User's Group
P.O. Box 37127
Albuquerque, NM 87176
(505) 884-3778

Commodore User's Group
Danny Byrne
6209 Leslie Place
Albuquerque, NM 87109
(505) 821-5812

High Plains Commodore Club
Don Joslin
Suite 1, Box 107
516 W. 21st Street
Clovis, NM 88101

Southern New Mexico Commodore
User's Group
P.O. Box 4437
Univ. Park Branch
Las Cruces, NM 88001
(505) 522-7622

New York

Manhattan 64
Greg Pumroy
139 Guernsey Street #1R
Brooklyn, NY 11222

Queens Commodore Users Group
Sam Soltan
67-42 Harrow Street
Forest Hills, NY 11375

Long Island VIC Enthusiasts
Arnold Friedman
17 Picadilly Road
Great Neck, NY 11023

Upstate Commodore Users Group
Chris Johnson
P.O. Box 5242 Arnot Mall
Horseheads, NY 14844

Vic/64 User Group
John Mason
249 S. 14th Street
Lindenhurst, NY 11757

Metropolitan Life Insurance Co.
Commodore Users Group
Alexander A. Priest
One Madison Avenue (13-W)
New York, NY 10010

New York Commodore User's Group
Shelly Roberts
135 Charles Street, Apt. 4-E
New York, NY 10022

"CitiiGroup"
N.Y. Vic 20/C-64 User Group
C/O N.Y. Amateur Computer Club
Attn: Brian Glover
P.O. Box 106
Church Street Station
New York, NY 10008

Oswego Commodore Users Group
Dr. John Boronkay
Dept. Industrial Arts & Technology
SUNY
Oswego, NY 13321

Schenectady Commodore Users Group
Bill Pytlovany
7 Center Street
Scotia, NY 12302

North Carolina

Commodore Users Group of Fayetteville
Steve Peterson
612 Brittany Place
Fayetteville, NC 28304

Hendersonville Users Group
Jerry Baldwin
Rt. 6 — Box 101
Hendersonville, NC 28739

North Dakota

Capitol City Computer Club
Rolf Arnold
c/o Veterans Mem. Library
520 Avenue A East
Bismarck, ND 58501

Ohio

Commodore Users in Akron
P.O. Box 9243
Akron, OH 44305

Clercom 64
Wally Jones
5152 Benton Road
Batavia, OH 45103

CPU Connection
P.O. Box 42032
Brook Park, OH 44142

Commodore Users of Blue Chip
Ted Stalets
816 Beecher Street
Cincinnati, OH 45206
(513) 961-6582

Southwestern Ohio Commodore User's Group
P.O. Box 399117
Cincinnati, OH 45239

Central Ohio Commodore Users Group
Philip Lynch
P.O. Box 28229
Columbus, OH 43228-0229

Commodore Local Users Exchange (C.L.U.E.)
Pat Murphy
3040 Highcliff Court
Columbus, OH 43229

Akron Area C-64 User's Group
2453 Second Street
Cuyahoga Falls, OH 44221
(216) 923-4396

C-64 Users of Philadelphia
Bonnie Mackie
1292 Bluebird, SW
Dellroy, OH 44620

CPU Connection
Bob Zak
10201 Plymouth
Garfield Heights, OH 44125

SE Cleveland Commodore Crazies
Carl Skala
18813 Harlan Drive
Maple Heights, OH 44137
(216) 581-3099

Medina Commodore Users Group
Jill Carpenter
P.O. Box 182
Medina, OH 44258
(216) 722-2611

Commodore 64 User's Group
702 Park Ave. NW
New Philadelphia, OH 44663

Cincinnati Commodore Computer Club
Attn: Wally Jones
P.O. Box 450
Owensville, OH 45160

Commodore Computer Club of Toledo
P.O. Box 8909
Toledo, OH 43623

Dayton Area Commodore Users Group
Charles Tobin
679 Murray Hill Drive
Xenia, OH 45385
(513) 372-4077

Oklahoma

Commodore Users of Bartlesville
Fred Mayes
1704 South Osage
Bartlesville, OK 74003
(918) 336-0233

Southwest Oklahoma Computer Club
c/o Commodore Chapter
P.O. Box 6646
Lawton, OK 73504

Commodore User's Group
c/o Muskogee Computer Soc.
202 South 12th Street
Muskogee, OK 74401

Commodore Users of Norman
209 Brookwood
Noble, OK 73068

Commodore Users
P.O. Box 268
Oklahoma City, OK 73101

Commodore Oklahoma User's Club
4000 N.W. 14th Street
Oklahoma City, OK 73107
(405) 943-1370

Tulsa Area Commodore User's Group
7804 N. 117th E. Avenue
Owasso, OK 74055
(918) 272-9755

Oregon

Albany/Corvallis 64 Users Group
Al Rasmus: (503) 967-8701
Steve Johnson: (503) 752-7614

Astoria Commodore Users Group
Kent Paulsen: (503) 325-5685

Beaverton 64 Users Group
Jack Simpson
(503) 646-9596

Coos Computer Club
P. L. Chard, Secretary
P.O. Box 4066
Eastside, OR 97420

Lane County C-64 Users Group
P.O. Box 11316
Eugene, OR 97440
(503) 726-2131

Springfield Commodore Users Group
4400 Franklin Ave., Suite 1443
Eugene, OR 97403
(503) 741-2522

Southern Oregon VIC/C-64 User's Group
3600 Madrona Lane
Medford, OR 97501
(503) 779-7631

Milwaukie Commodore Users Group
Mike Garvin: (503) 255-1376

Mt. Hood Community College Users Group
Bart Snell
(503) 667-7078

Southern Oregon VIC/64 Users Group
P.O. Box 765
Phoenix, OR 97501

Oregon Commodore Computer Users Group
Howard MacMurray, Secretary
3636 S.W. Patton Road
Portland, OR 97221
(503) 224-6707

North Portland Commodore Users Group
VIC-20 SIG
Doug McMurdo: (503) 245-2571
C-64 SIG
Gary Thompson: (503) 666-4131

United States Commodore Users Group
Richard Tsukiji
P.O. Box 2310
Roseburg, OR 97470
(503) 672-7591

Salem 64 Users Group
3795 Saxon Drive S.
Salem, OR 97302

Yamhill County Commodore Users
Route 2, Box 246
Sheridan, OR 97378

Columbia County Commodore Club
Gary Thomas
424 South 17th Street
St. Helens, OR 97051

Columbia Commodore Computer Users
Ken Denniston
1214 Washington
The Dalles, OR 97058
(503) 298-5965

Pennsylvania

Lehigh Valley Commodore User Group
Hiram Cook
2228 Baker Drive
Allentown, PA 18102

Ft. Indiantown Gap Commodore Club
William Fuerst
Bldgs 9-11, Mail & Distribution
Ft. Indiantown Gap
Annville, PA 17003

Scranton Commodore Users Group
Mark Davis
P.O. Box 211
Clarks Summit, PA 18411

NEPACC Commodaters
P.O. Box 644
Conyngham, PA 18219

North Coast Commodore Users Group
Dale Niedzielski
P.O. Box 493
Fairview, PA 16415

Westmoreland Computer Users Club
Commodore Section
Jim Mathers
3021 Ben Venue Drive
Greensburg, PA 15601

Capital Area Commodore Club
Brad Mitchell
1089 D Michigan Drive
Harrisburg, PA 17111

Indiana Commodore Users Group
Richard Halapin
98 Rex Ave.
Indiana, PA 15613

Wyoming Valley Commodore Users Group
John Youells
94 Second Ave.
Kingston, PA 18704

Eight Squared
Mindy Skelton
P.O. Box 76
Mt. Holly Springs, PA 17065

Sperry-New Holland Users Group
John Bowman
500 Diller Ave.
New Holland, PA 17557

Philadelphia Area Computer Society — CBM Section
Dennis Sosna
919 E Godfrey Ave.
Philadelphia, PA 15236

Pittsburgh Commodore Group
Joel Casar
2015 Garrick Drive
Pittsburgh, PA 15235

Main Line Commodore Users Group
Emil Volcheck
1046 General Allen Lane
West Chester, PA 19380

South Pittsburgh Commodore Users Group
Chuck Groves
2407 Pennsylvania Ave.
West Mifflin, PA 15122

West Branch Commodore Users Group
Gene Loveland
P.O. Box 995
Williamsport, PA 17703

Rhode Island

Commodore User's Group
978 Tiogue Avenue
Coventry, RI 02816
(401) 828-7385

Irving Silverman CPA
160 Taunton Avenue
E. Providence, RI 02914

Newport VIC/64 Users
10 Maitland Court
Newport, RI 02840
(401) 849-2684

Rhode Island Computer Enthusiasts (RICE)
Michael Skelton
198 Morris Avenue
Pawtucket, RI 02860
(401) 728-8602

South Carolina

Commodore Computer Club of Columbia
318 Quincannon Drive
Columbia, SC 29210

Computer User's Society of Greenville
347 S. Pleasantburg Drive
Greenville, SC 29607
(803) 235-7922

Charleston Computer Society
P.O. Box 5264
N. Charleston, SC 29406
(803) 747-0310

Spartanburg Commodore User's Group
803 Lucerne Drive
Spartanburg, SC 29302
(803) 582-5897

South Dakota

PET User's Group
515 South Duff
Mitchell, SD 57301
(605) 966-8277

VIC/64 User's Club
203 E. Sioux Avenue
Pierre, SD 57501
(605) 224-4863

Tennessee

Commodore User Club
Metro Computer Center
1800 Dayton Blvd.
Chattanooga, TN 37405

Commodore Computer Club
P.O. Box 96
Estill Springs, TN 37330
(615) 649-5962

ET 64 User's Group
P.O. Box 495
Knoxville, TN 37901

Metro-Knoxville Commodore User Club
7405 Oxmoor Road
Knoxville, TN 37931
(615) 938-3773

Memphis Commodore User's Group
2476 Redvers Avenue
Memphis, TN 38127

Nashville Commodore User's Group
P.O. Box 121282
Nashville, TN 37212

Texas

Commodore Computer Club of Austin
Roy Holmess/Dr. Jerry Frazee
P.O. Box 49138
Austin, TX 78765

Commodore User's Group
5326 Cameron Road
Austin, TX 78723
(512) 459-1220

SCOPE
1020 Summit Circle
Carrolton, TX 75006

Corpus Christi Commodores
Jim O'Rear / Bob McKelvy
P.O. Box 6541
Corpus Christi, TX 78411

Gulf Coast Commodore Users Group
Lawrence Hernandez
P.O. Box 128
Corpus Christi, TX 78403
(512) 887-4577

Corpus Christi Commodore Users
3650 Topeka Street
Corpus Christi, TX 78411

64 User's Group, Inc.
Stan Gordin
P.O. Box 801828
13604 Midway Road
Dallas, TX 75380

PET Users
2001 Bryan Tower, Suite 3800
Dallas, TX 75201

El Paso 64s
1713 Dean Martin Street
El Paso, TX 79936
(915) 855-1107

Tarant County Commodore 64 Club
Jeff Speed
1901 Lanewood
Ft. Worth, TX 76112

VIC 20 Users Group
Jeff Southerland
6416 Brookhaven Trail
Ft. Worth, TX 76133

Commodore Houston Users Group (CHUG)
John Walker
8738 Wildforest
Houston, TX 77088
(713) 999-3650

Savid Computer Club
Davi Jordan
312 West Alabama — Suite #2
Houston, TX 77006

Mid Cities Commodore Club
Garry Wordelman
413 Chisolm Trail
Hurst, TX 76053

Irving Commodore Users Group
Robert Hayes
3237 Northgate #1289
Irving, TX 75062
(214) 252-7017

The NASACOM 64 Commodore Club
610 Bayridge Road
La Porte, TX 77571
(713) 471-9622

Longview User's Group
P.O. Box 9284
Longview, TX 75606
(214) 777-4458
(214) 759-0699

South Plains 64 Users Group
John N. Bottoms
7709 Avenue 'W'
Lubbock, TX 79423
(806) 745-4381

VIC Users Group
3817 64th Street
Lubbock, TX 79413

Commodore Users Group
James Meeker
1110 Texas Avenue
Mart, TX 76664

BiStone Users Club
R. G. Gore
P.O. Box 386
Mexia, TX 76667

Commodore User's Group of Odessa
2904 N. Alleghaney
Odessa, TX 79764
(915) 332-2582

Commodore Computer Club (C3)
Randy Mills
2217 North Sumner
Pampa, TX 79065

64 Users Group
S. G. Grodin
2421 Midnight Circle
Plano, TX 75075

SCOPE
P.O. Box 3095
Richardson, TX 75083

Commodore Users Group
Larry Williams
P.O. Box 652
San Antonio, TX 78293

Interface Computer Club
M. E. Garza
814 North Sabinas
San Antonio, TX 78207

The Great Northwest CBM 64 Users Group
Attn: Randy
6302 War Hawk Drive
San Antonio, TX 78238

Commodore Users Group
Danny Miller
624 Bellview Street
Sulphur Springs, TX 75482

PET User Group
Texas A&M Microcomputer Club
Texas A&M, TX 77843

The Woodlands Commodore Users Group
Andrew Gardner
3 Splitrock Road
The Woodlands, TX 77380

Crossroads Commodore Users Group
Jerry Guy
417 Irma Drive
Victoria, TX 77901
(512) 575-0342

Computas
Burl A. Hays
Route 4, Box 214
Waco, TX 76705

Utah

The Commodore User's Group
652 West 700 North
Clearfield, UT 84015
(801) 776-3950

Northern Utah VIC & 64
P.O. Box 533
Garland, UT 84312

UVCS Utah Valley Computer Society
330 North Canal Drive
Lindon, UT 84062
(801) 785-5467

Utah PUG
2236 Washington Blvd.
Ogden, UT 84401

The Commodore User's Club
742 Taylor Avenue
Ogden, UT 84404

Uintah Basin Commodore User's Club
P.O. Box 1102
Roosevelt, UT 84066

Mountain Computer Society
P.O. Box 1154
Sandy, UT 84071
(801) 571-6813

Vermont

Burlington Area Commodore User's Group
6 Mayfair
South Burlington, VT 05041
(802) 658-4160

Virginia

Franconia Commodore Users Group
Mark Sowash
6209 Rose Hill Drive
Alexandria, VA 22310

Navair Commodore Users Group
Jon Steeby
4423 Raleigh Ave. #102
Alexandria, VA 22304

Arlington Victims (20/64)
Clifton Gladney
4501 Arlington Blvd.
Arlington, VA 22204

Dale City Commodore Users Group
Pat Sullivan
4303 Hemingway Drive
Dale City, VA 22193

Piedmont Users Group
David Gray
135 Beverly Road
Danville, VA 24514

PENTAF (Pentagon)
Ralph Poole
9912 Colony Road
Fairfax, VA 22030

Franklin Users Group
Bruce Powell
1201 North High Street
Franklin, VA 23815

Fredricksburg CCC
Steve Northcutt
Box 1011 College Station
Fredricksburg, VA 22402

Washington Area C-64 User Group
Martin Smith
7426 Eldorado Street
Mclean, VA 22012

Washington Area C-64 User Group
Dick Jackson
P.O. Box 93
Mt. Vernon, VA 22121

Tidewater Commodore Users
P.O. Box 61814
Virginia Beach, VA 23462

Washington

C-64 Diversity
Jill Johnston
18204 67th Avenue
Arlington, WA 98223

CBM User's Group
c/o Rick Beaber
803 Euclid Way
Centralia, WA 98531
(206) 736-4085

Edmonds Commodore Users Group
Rick Ellis
(206) 776-3539

Fort Lewis Commodore Computer Club
c/o Jim Litchfield
Quarters 2821-A
Fort Lewis, WA 98433

Longview Commodore Users Group
Dan Davis
(206) 577-8224

Arlington Commodore User's Group
Jane M. King, President
4416-126th Place N.E.
Marysville, WA 98270
(206) 653-3882

Bunch 'a' Bytes
4916 121st Place, N.E.
Marysville, WA 98270

Whidby Island Commodore Computer Club
P.O. Box 1471
Oak Harbor, WA 98277
(206) 675-8535

Tri-Cities Commodore Computer Club
1926 Pine Street
Richland, WA 99352
(509) 943-4734

PET Users Group
1800 Taylor Ave. N102
Seattle, WA 98102

NW PET Users Group
2565 Dexter N. 3203
Seattle, WA 98109

Computer Club
c/o Honeywell, Inc.
5303 Shilshole Avenue, N.W.
Seattle, WA 98107
(206) 789-2000 Ext. 1402

East Seattle Commodore Users Group
Erik Moseid
(206) 883-7404

Spokane Commodore User Group
N. 4311 Whitehouse
Spokane, WA 99205
(509) 328-1464

World Wide User Group
R. Smith
P.O. Box 98682
Tacoma, WA 98498
24-Hr. BBS: (206) 535-0574

Vancouver VIC/64 Users Group
Bruce Qualman
(206) 256-7206

Blue Mountain Commodore Users
Jim Godfrey, Secretary
550 South 2nd Avenue
Walla Walla, WA 99362-3149
(509) 529-4663

Central Washington Users Group
1222 S. First Street
Yakima, WA 98902

West Virginia

Personal Computer Club
P.O. Box 1301
Charleston, WV 25325

Tri-State Commodore Users
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124

Commodore Home User's Group
81 Lynwood Avenue
Wheeling, WV 26003
(304) 242-8362
(304) 242-2605

Wisconsin

Club 64
6156 Douglas Avenue
Caledonia, WI 53108
(414) 835-4645

Eau Claire CBM64 User's Group
Rt. 5, Box 179 A
Eau Claire, WI 54703
(715) 874-5972

Milwaukee Area CBM64 Enthusiasts
P.O. Box 340
Elm Grove, WI 53122
(414) 259-5991

Commodore User Group
1130 Elm Grove Street
Elm Grove, WI 53122

Comm Bay 64
Jeff Schwecler
2589 Haven Road
Green Bay, WI 54303
(414) 439-1619

COMAL User's Group
5501 Groveland Terrace
Madison, WI 53716
(608) 222-4432

Menomonie Area Commodore User's Group
510 12th Street
Menomonie, WI 54751
(715) 235-4987

SWITCH
W. 156 N. 8834 Pilgrim Road
Menomonie Falls, WI 53051
(414) 255-7044

Madison Area Commodore User's Group
1552 Park Street
Middleton, WI 53562
(608) 831-4852

2001 64 and VIC User's Group
3119 North 44th Street
Milwaukee, WI 53216
(414) 445-2117

VIC 20 & C-64 User Group
522 West Bergen Drive
Milwaukee, WI 53217
(414) 476-8125

SEWPUS
P.O. Box 21851
Milwaukee, WI 53221

Fond Du Lac Area Commodore User's Group
1504 Shelley Court
North Fond Du Lac, WI 54935

Coulee Country Commodore Club
W. 6581 Oak Park Drive
Onalaska, WI 54650

Commodore 64 Software Exchange
P.O. Box 224
Oregon, WI 53575

CUSSH
3614 Sovereign Drive
Racine, WI 53406
(414) 554-0156

Wisconsin Association of VIC/C-64 Enthusiasts
P.O. Box 641
Waukesha, WI 53187-0641
(414) 771-7016
(414) 964-3704

Waukesha Area Commodore
256 ½ West Broadway
Waukesha, WI 53186
(414) 547-9391

CHIPS
1017 Kilbourn Avenue
West Bend, WI 53095
(414) 334-2494

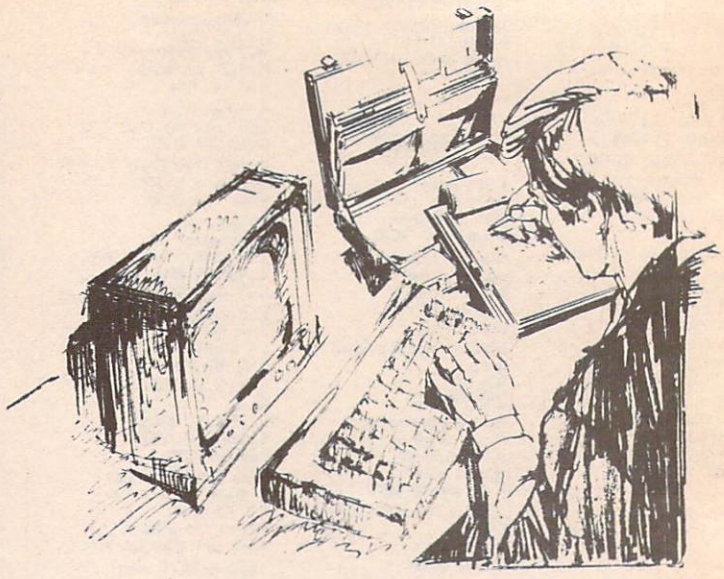
Wyoming

Commodore User's Group
670 North Third, #B
Laramie, WY 82070
(307) 721-5908

CANADA

Toronto PET Users Group (TPUG)
Membership Dept. A3
101 Duncan Mill Road — Suite G7
Don Mills, Ontario M3B 1Z3
(416) 445-4524

How To Type In Program Listings From The Guide



In order to typeset programs so that clear images may be printed in the pages of *The Guide*, it was necessary to deal with the problem of graphics characters that appear on the screen when you type in a capital letter in graphics mode, or when you choose graphic symbols for colors (instead of using POKEs, which occupy more memory space in your programs), etc.

To begin with, all programs appear in the text mode. You enter the text mode by pressing the Commodore key and the shift key simultaneously. This solves the problem of capital letters.

The other graphic symbols are replaced with letters the typesetting machine can recognize. For example, if the program shows [lt grn], you simultaneously press the Commodore key and 6, causing the graphic symbol for light green to be shown on your screen.

We hope this helps clear up any confusion you may have experienced. If you have any questions, please feel free to contact us. Have fun!

Program Shows:	Press Keys:	Screen Shows:
[blk]	ctrl-1	
[wht]	ctrl-2	
[red]	ctrl-3	
[cyn]	ctrl-4	
[pur]	ctrl-5	
[grn]	ctrl-6	
[blu]	ctrl-7	
[yel]	ctrl-8	
[rvs on]	ctrl-9	
[rvs off]	ctrl-0	
[orange]	Cmdr-1	
[brown]	Cmdr-2	
[lt red]	Cmdr-3	
[gray 1]	Cmdr-4	
[gray 2]	Cmdr-5	
[lt grn]	Cmdr-6	
[lt blu]	Cmdr-7	
[gray 3]	Cmdr-8	
[clr]	Shft-Clr	
[home]	Home	
[up]	Crsr-Up	
[dwn]	Crsr-Down	
[left]	Crsr-Left	
[right]	Crsr-Right	
[f1]	f1	
[f3]	f3	
[f5]	f5	
[f7]	f7	
[up-arrow]	Up Arrow	

NEXT MONTH IN THE GUIDE

☐ **The Tube:**

A look behind the screen at the inner workings of your monitor, examining both the mechanics and history of the most visual part of your computer system.

☐ **Leader Board:**

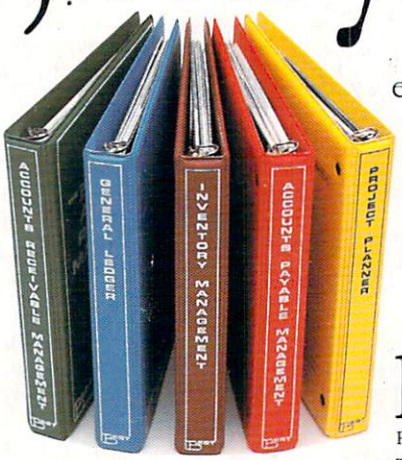
Golf simulator sets new graphic standards. Join Randy next issue for an entertaining round of golf.

☐ **The Multi-faceted Amiga:**
Grant takes you further into the many windowed world of the Amiga, exploring and explaining multi-tasking.

Simple, easy to use.

Professional quality defines new B.E.S.T. software business management systems for the Commodore 64/128.* Ease of use is unique. Manuals are illustrated, readable and easy to understand. Menus are clear and concise.

B.E.S.T. software gives you more time to focus on business. (No need for computer or accounting experience.) B.E.S.T. systems *seem* simple. Yet this is very sophisticated software, easily capable of becoming the indispensible management tool.



Our objective is to make your business easier to manage.

Available at your local computer store.

Prices: Payables & Receivables, $59.95 each; Inventory & Ledger $69.95 each; Project Planner, $89.95.

*Trademark of Commodore Business Machines, Inc.

B.E.S.T. Business Electronics Software & Technology, Inc.
P.O. Box 852 / McMinnville, Oregon 97128 / (503) 472-9512
Toll Free 1-800/368-BEST